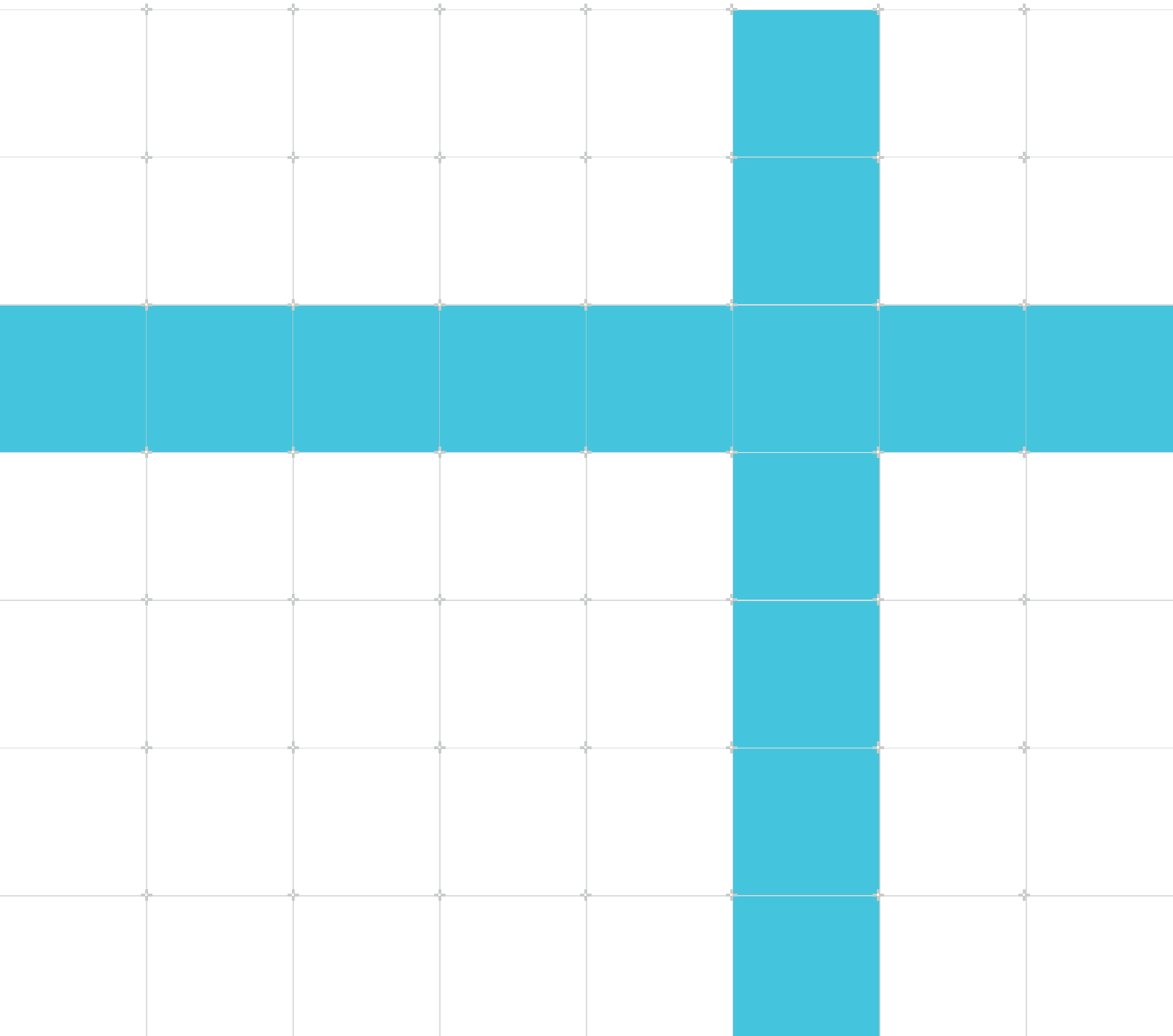
Arm® Neoverse™ V3 Core

Telemetry Specification

Non-Confidential
Copyright © 2024–2025 Arm Limited (or its affiliates).
All rights reserved.

Issue 02
107905_0200_02_en



Arm® Neoverse™ V3 Core Telemetry Specification

This document is Non-Confidential.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.
Arm only permits use of this document if you have reviewed and accepted Arm's Proprietary Notice found at the end of this document.

This document (107905_0200_02_en) was issued on 2025-08-15. There might be a later issue at https://developer.arm.com/documentation/107905

See also: Proprietary notice | Product and document information | Useful resources

Start reading

If you prefer, you can skip to the start of the content.

Intended audience

This specification is useful for engineers to collect and analyze Arm® Neoverse™ V3 core telemetry data to gain insights about a system's performance. Architects and system designers can also use it for resource characterization and platform tuning.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com.

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Contents

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

1. Overview of the Neoverse V3 core Telemetry methodology

The *Arm® Neoverse V3 Core Telemetry Specification* describes the Topdown methodology, derived metrics, and Performance Monitoring Unit (PMU) events supported by the Arm Neoverse V3 core, also known as the processor.

Note

This specification is applicable to all releases of the product.

This specification implements the framework provided by the *Arm® CPU Telemetry Solution Topdown Methodology Specification*, which is referred to as the Architecture Specification. The reader is expected to read this document in conjunction with the Architecture Specification.

Arm Telemetry framework

This specification outlines the telemetry features implemented for the Arm Neoverse V3 core and follows the Arm Telemetry framework for CPUs defined in the Architecture Specification.

The following list provides a brief description of the Telemetry framework:

Events

Hardware performance monitoring events implemented by the core that contain raw data read from the registers or memory buffers.

Metrics

Derived mathematical relationships between events that provide insight into the system behavior. They are developed to abstract hardware details of the events from consumers of the telemetry data.

Metric groups

Group of metrics that can be analyzed together to investigate a bottleneck scenario or a specific resource in a given system.

Methodology

Actionable guidance, such as Arm Topdown methodology, to explain how to consume the different metrics and events for a specific usage model. Decision tree with a group of metrics that can be analyzed hierarchically to investigate a bottleneck scenario or a specific resource in a given system.

Tool support for profiling and monitoring

This specification is also available in a machine-readable format (JSON) to be consumed by profiling and monitoring tools. The JSON schema implements the Arm Telemetry framework from the Architecture Specification.

The JSON for the core is published in the open source *Arm® Telemetry Solution GitLab repository*.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

1.1 Documentation and resources

Arm products include a set of documents.

The documentation and resources for Neoverse V3 consist of:

- *Arm® Telemetry on Arm Developer*
- *Arm® Telemetry Solution GitLab repository*
- *Arm® Neoverse™ V3 Core Technical Reference Manual*

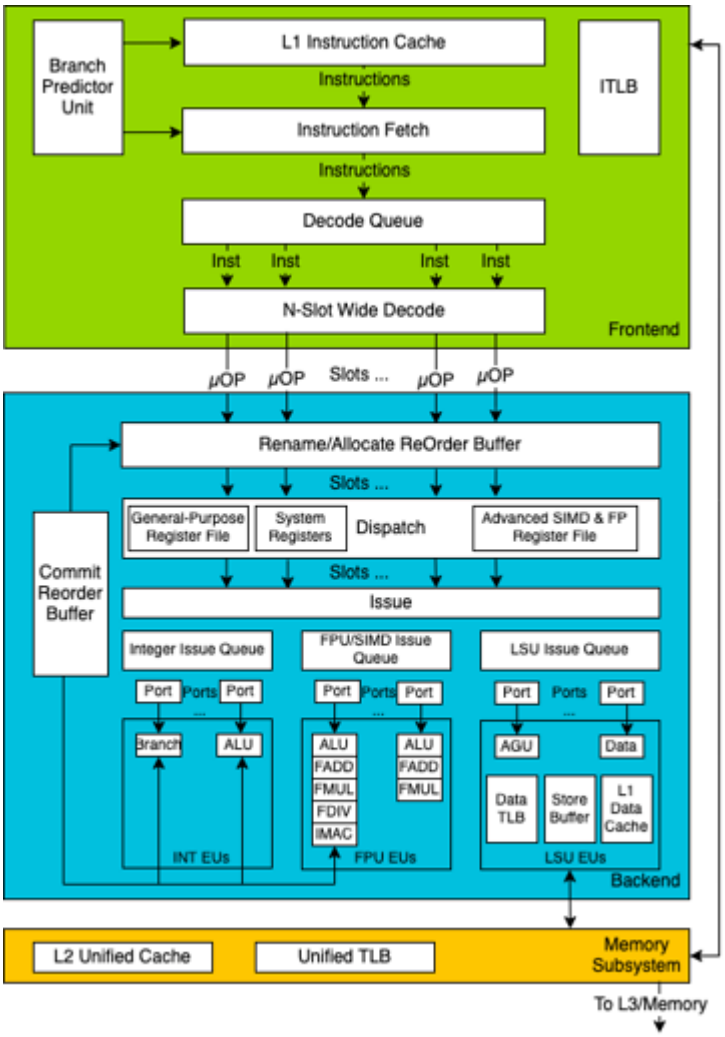
Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2. Telemetry features of Neoverse V3 core

The Neoverse V3 core is a super pipelined superscalar processor that has an in-order frontend and out-of-order backend.

The following figure shows the microarchitecture details of the Neoverse V3 core.

Figure 2-1: Neoverse V3 core microarchitecture



The frontend of the core pipeline is comprised of the instruction fetch and decode units. The frontend also includes a branch predictor unit that predicts branch target addresses and fetches instructions ahead of the pipeline. This unit helps to hide latencies caused by control flow bubbles in the pipeline. The fetch unit can fetch multiple instructions for each cycle whose bandwidth is specific to a microarchitecture design, which gets stored in a decode queue. The decode queue sends multiple instructions per cycle for decoding, whose bandwidth is determined by the number of decode slots available. The decode unit decomposes the Arm architecture instructions into micro-operations, also known as micro-ops or (µops). This unit decodes more than one micro-

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

operation for each cycle, which are then fed to the rename unit for organization for out-of-order execution in the backend of the core. The instruction bandwidth is determined by the number of renamed slots available in the microarchitecture. The number of slots can be discovered in the SLOTS field of the PMMIR_EL1 registers. From a microarchitecture standpoint, the rename unit is considered the boundary between the frontend and backend of the core.

The backend of the core has a scheduler that orchestrates the operations to be executed when the issue queue associated with the operation can accept the operation. The issue queue sends operations for execution when the execution unit is free and the source operands are ready. Once the execution is complete, the results are sent to the commit Reorder Buffer (ROB) from where the instructions are retired when the speculated execution is confirmed. The backend of the core executes the operations out-of-order and stores results with the help of the reorder buffer. The dispatch unit tracks dependencies between operations and determines the operand availability for the execution of operations. Register renaming is occurs at this stage to mitigate data dependency hazards.

In the dispatch unit, issue queues are employed for:

- Queuing the micro-operations (μops) to assigned ports
- Managing dependencies between operations
- Tracking operand availability for execution

Each execution port supports different categories of operations. After the execution of operations, the ROB is updated with execution results. Completed operations are retired architecturally in the right program order. Operations are flushed when the predicted program flow changes due to mispredictions or exceptions. The following execution units (EU) support typical operation types:

- Integer EU executes branches and arithmetic instructions.
- Floating-Point Unit (FPU)/Single Instruction Multiple Data (SIMD) EU executes the FP instruction and vector instructions.
- Load Store Unit executes load/store/atomic instructions.

The Memory subsystem of the core handles the execution of load and store operations which rely heavily on the memory hierarchy levels. The Neoverse V3 core has dedicated cache levels, L1 and L2 for each core, where the L2 cache is shared between the L1 data cache and the L1 instruction cache. The Load Store Unit controls the data flow between the caches and to memory. The core has multiple load/store units, which can handle both read and write operations. L1 and L2 caches are set-associative. The size of the cache is configurable for each implementation and determines the number of sets in each way. The private L2 cache of the core connects to the rest of the system through an AMBA® 5 CHI interface.

Neoverse V3 system configurations

All systems with the Arm® Neoverse™ Coherent Mesh Network support a shared system-level cache. Understanding the cache hierarchy and configuration of the system being analyzed is crucial in deriving insights from the cache effectiveness Performance Monitoring Unit (PMU) events.

It is always best to check with the Silicon Provider for details on the system configuration for the underlying system, including the cache sizes.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

PMU capabilities of Neoverse V3

The Neoverse V3 core implements version 3.7 of the Performance Monitors Extension, `FEAT_PMUv3p7`, and Arm v8.4 debug architecture, `FEAT_Debugv8p4`.

For more information, see *Arm® Architecture Reference Manual for A-profile architecture*.

The Neoverse V3 PMU has six configurable counter registers and one dedicated function counter to count CPU cycles.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3. CPU performance analysis methodology

The Arm Topdown methodology for the Neoverse V3 enables you to use PMU events, metrics, and metric groups to identify potential bottlenecks that could negatively impact the performance of the core in your design.

The methodology is conducted in two stages.

Stage 1: Topdown analysis

The first stage is to perform Topdown analysis to detect and identify any performance bottlenecks in the CPU, and provide direction for further analysis at Stage 2. Stage 1 uses hierarchical pipeline stall-related metrics. For more information, see Stage 1: Topdown analysis.

Stage 2: Microarchitecture Exploration

The second stage is to conduct microarchitecture exploration to further analyze bottlenecked CPU resources, based on the Stage 1 findings. Stage 2 uses a set of CPU resource-effectiveness metrics. For more information, see Stage 2: Microarchitecture Exploration.

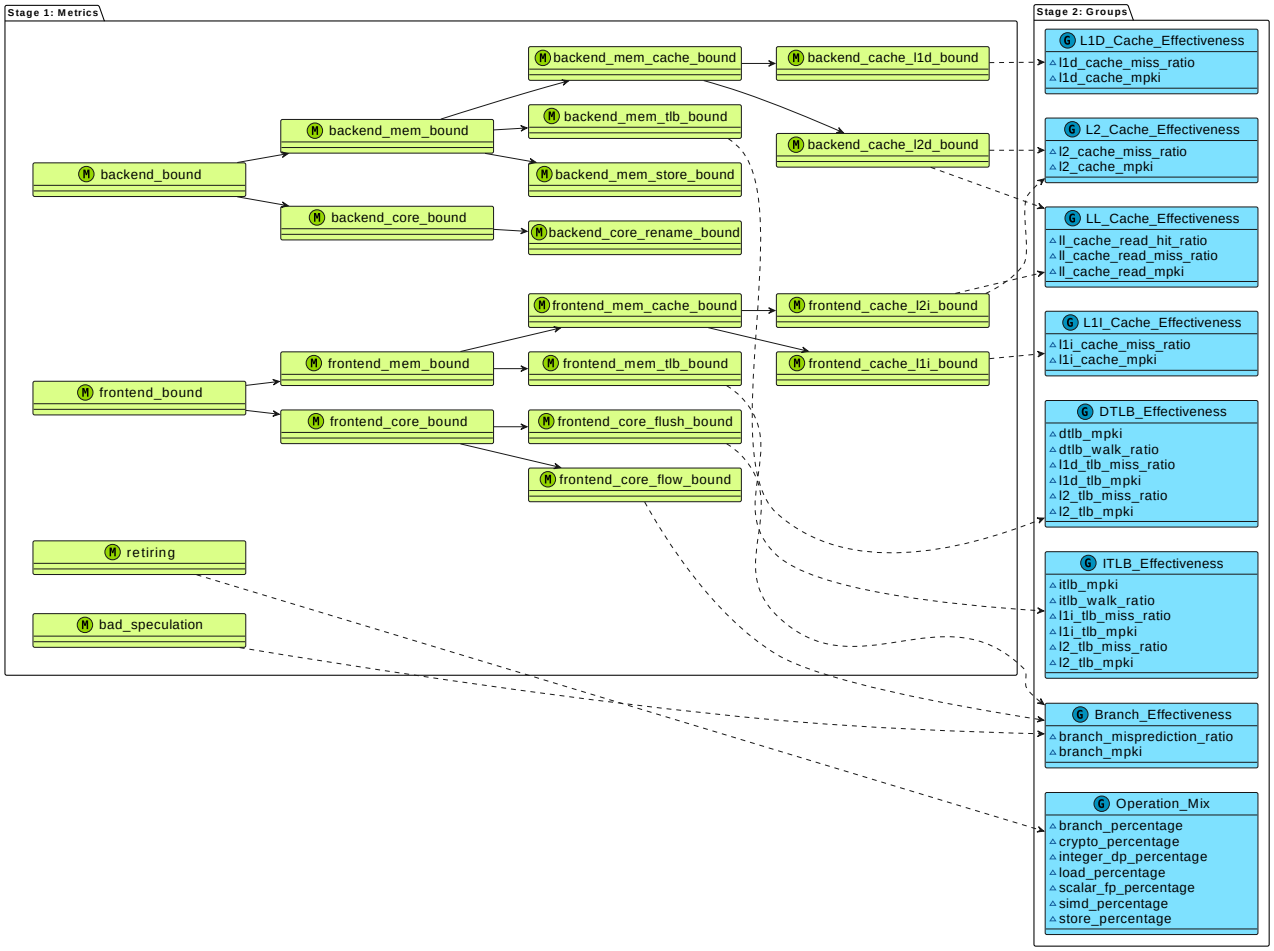
For more information about our approach to performance analysis and the standardized telemetry framework, see *Arm® CPU Telemetry Solution Topdown Methodology Specification*.

We recommend collecting all metrics that are in Stage 1 and Stage 2 Topdown analysis for workload characterization. We provide a recommended set of microarchitecture exploration metric groups for further analysis, for hotspots detected in Stage 1. All Stage 2 metrics can be used to derive further insights into the overall microarchitecture behavior during the execution of the application under investigation. These metrics can be used independently of Stage 1.

The following figure gives an overview of the Topdown methodology tree for Neoverse V3. It shows Stage 1 metrics, and Stage 2 metric groups and metrics. Stage 1 covers the stall-related metrics for Topdown analysis for bottleneck identification. Stage 2 covers the microarchitecture exploration metric groups for root cause analysis.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-1: Topdown methodology overview for Neoverse V3



Note

The industry-standard metrics Misses Per Kilo Instructions (MPKI) and Miss Ratios are metric groups that are defined in Stage 2, but they are not included in the decision tree.

3.1 Stage 1: Topdown analysis

The objective of the Topdown analysis is to identify potential bottlenecks in the key blocks of the processor. Each block can be further broken down to units and subunits within the block that all have different performance characteristics. For example, control flow and data flow issues can be fixed differently in software after root cause analysis.

Results from a Topdown analysis can indicate:

- Which metric groups and metrics to analyze next.
- Areas where software improvements can be made in the current design.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Existing microarchitectural limitations that can inform future hardware improvements, better system configuration, or tuning decisions at a platform level.

In Neoverse V3 there are three Stage 1 metric groups.

Topdown Level 1

The Topdown Level 1 metric group contains the first set of metrics to begin Topdown analysis of application performance. These metrics provide the percentage distribution of processor pipeline utilization.

For more information about the metrics in this group and the associated formulas and events, see Topdown_L1.

Topdown Frontend

The Topdown Frontend metric group contains a set of metrics to analyze a frontend bound workload.

For more information about the metrics in this group and the associated formulas and events, see Topdown_Frontend.

Topdown Backend

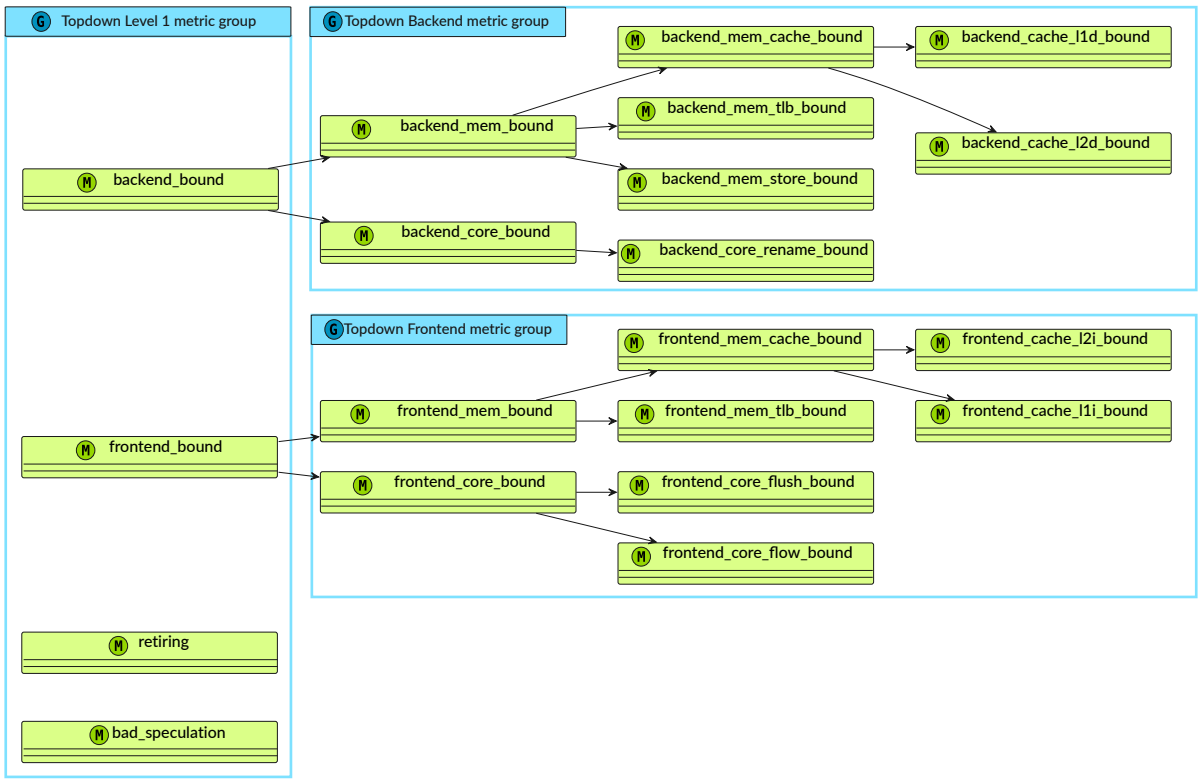
The Topdown Backend metric group contains a set of metrics to analyze a backend bound workload.

For more information about the metrics in this group and the associated formulas and events, see Topdown_Backend.

The following figure shows the metric groups and metrics in the Stage 1 Topdown methodology tree for Neoverse V3, which supports up to four levels of hierarchical pipeline stall accounting.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-2: Neoverse V3 Topdown methodology Stage 1 overview



3.1.1 Topdown Level 1 metric group

The Topdown_L1 metric group contains the first set of metrics to begin Topdown analysis of application performance. These metrics provide the percentage distribution of processor pipeline utilization.

The following implementation criteria apply for this metric group:

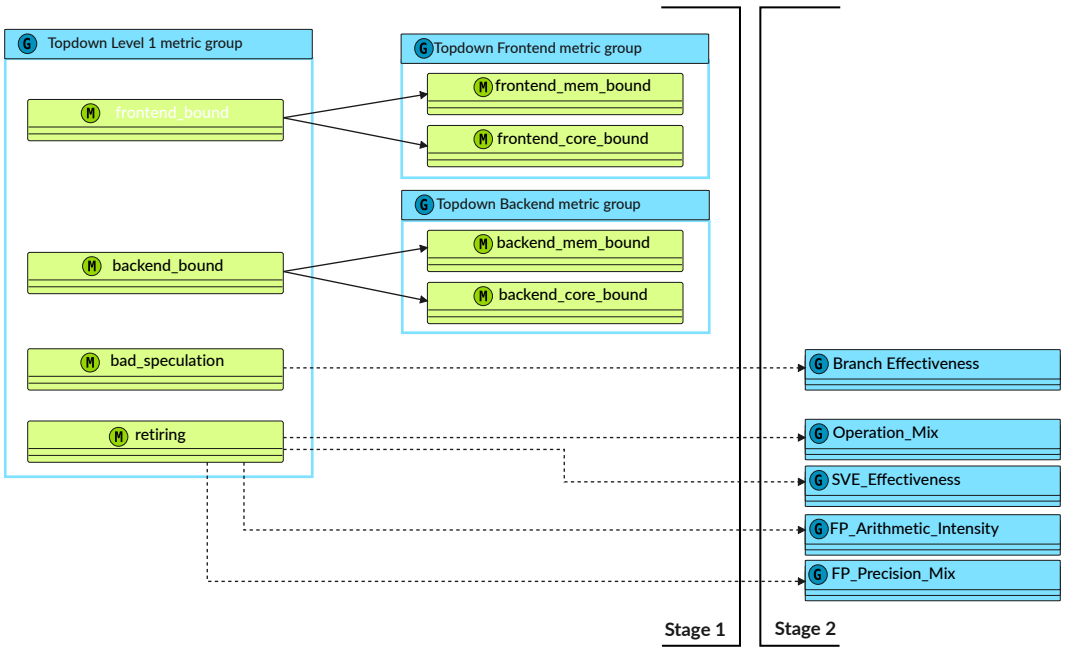
- The sum of the metrics in Topdown_L1 equals 100% of the total execution cycles in the core implementation.
- The total execution bandwidth of the core is the same as the number of execution slots for operations.
- Slots are the number of ports in the rename unit, which is considered the boundary between the frontend and backend of the core.
- A frontend stall is counted when there are no micro-ops to rename.
- A backend stall is counted when there are micro-ops in the rename unit, but they cannot dispatch to the backend due to backend resource constraints.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- In a cycle where there are no micro-ops to rename and no micro-ops dispatched to the backend, the CPU counts a frontend stall.
- The definition of the backend stall is linked to the slot that does not dispatch although there are micro-ops available in the rename unit.

The following figure shows the metrics for Topdown_L1 and the next steps in the methodology.

Figure 3-3: Neoverse V3 Metric Group: Topdown_L1 and next steps



The metrics in the Topdown Level 1 metric group are:

frontend_bound

The frontend_bound metric measures pipeline inefficiency due to slots that were stalled because of resource constraints in the frontend.

For further analysis, the next step is the Stage 2 Topdown Frontend metric group.

backend_bound

The backend_bound metric measures pipeline inefficiency due to slots that were stalled because of resource constraints in the backend.

For further analysis, the next step is the Stage 2 Topdown Backend metric group.

bad_speculation

The bad_speculation metric is the percentage of total slots that executed operations and did not retire due to a pipeline flush, which indicates cycles that were utilized inefficiently. This metric measures pipeline inefficiency caused by flushes in the pipeline, due to branch

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

mispredictions or machine clears. For workloads that demonstrate high bad speculation rates, the next step is to understand the branch performance in the workload.

Some key reasons for pipeline inefficiency are the same as the reasons that are identified in the `frontend_core_flush_bound` metric, which is part of the `frontend_core_bound` metric within the Topdown Frontend metric group.

For further analysis, the next step is the Stage 2 Branch Effectiveness metric group.

retiring

The retiring metric is the percentage of total slots that retired operations, which indicates cycles that were utilized efficiently. This metric covers the total slots that were utilized efficiently by the processor. A high retirement rate can also mean that there is an opportunity for further performance optimization.

Scalar code that shows high retirement can be optimized by vectorization if there is data parallelism. In some cases, the code that is executed can be made redundant or optimized. When you analyze high retirement rates, evaluation of the instruction or operation mix is a recommended next step in the characterization of execution units that are heavily utilized.

For further analysis, the next steps are Stage 2 Operation Mix metric group, Stage 2 Floating Point Arithmetic Intensity metric group, Stage 2 Floating Point Precision metric group, and Stage 2 SVE Effectiveness metric group.

3.1.2 Topdown Frontend metric group

The Topdown_Frontend metric group contains a set of metrics to analyze a frontend bound workload.

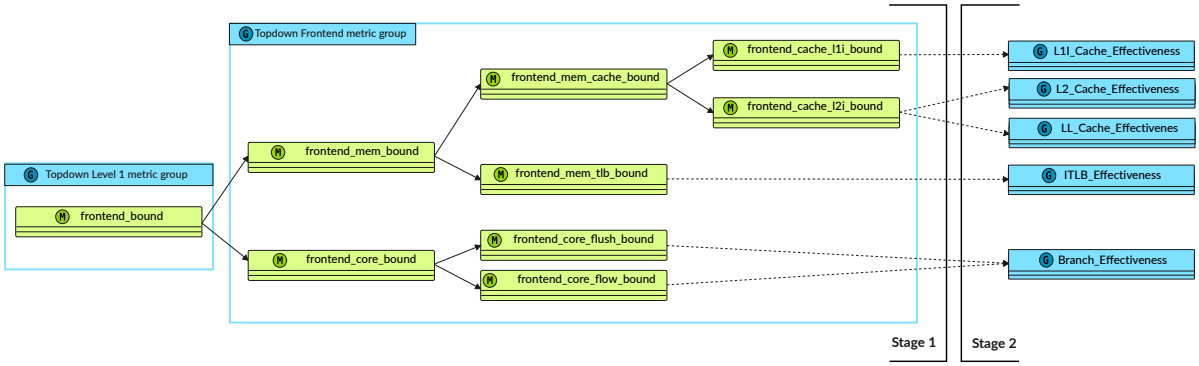
The following implementation criteria apply for this metric group:

- The implementation of the Frontend Core Bound metric in Neoverse V3 is the difference between the Frontend Bound and Frontend Memory Bound metrics. This means that the cycle was stalled in the frontend and the stall was not caused by the frontend memory units. This derivation is mathematically correct but misses a direct count of the Frontend Core Bound metric from an implementation standpoint.
- Due to the implementation, the sum of the Frontend Core Bound and Frontend Memory Bound metrics is equal to 100% of the Frontend Bound metric.

The following figure shows the metrics for the Topdown_Frontend metric group and the next steps in the methodology.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-4: Neoverse V3 Metric Group: Topdown_Frontend and next steps



The metrics in the Topdown Frontend metric group are:

frontend_mem_bound

The frontend_mem_bound metric is the percentage of total cycles stalled in the frontend due to frontend core resource constraints related to the instruction fetch latency issues caused by memory access components.

This metric breaks down into the following metrics:

- frontend_mem_cache_bound
  - frontend_cache_l1i_bound
  - frontend_cache_l2i_bound
- frontend_mem_tlb_bound

The following implementation criteria apply for these metrics:

- The Frontend Memory Cache Bound metric is implemented as the number of cycles of either the Frontend Memory L1I Bound metric or the Frontend Memory L2I Bound metric.
- The Frontend Memory L1I Bound metric counts when waiting on an L1 miss but not waiting on an L2 miss.

`frontend_mem_*_bound` stall metrics are derived ratios of Translation Lookaside Buffer (TLB) and Cache stalls such as:

- `frontend_mem_l1i_bound` is a ratio of frontend memory bound cycles waiting on an L1 miss.
- `frontend_mem_l2i_bound` is a ratio of frontend memory bound cycles waiting on an L2 miss.
- `frontend_mem_tlb_bound` is a ratio of frontend memory bound cycles waiting on a TLB miss.

For further analysis:

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- The next step for `frontend_cache_l1i_bound` is the Stage 2 Level 1 Instruction Cache Effectiveness metric group.
- The next step for `frontend_cache_l2i_bound` is the Stage 2 Level 2 Unified Cache Effectiveness metric group and the Stage 2 Last Level Cache Effectiveness metric group.
- The next step for `frontend_mem_tlb_bound` is the Stage 2 Instruction TLB Effectiveness metric group.

frontend_core_bound

The frontend_core_bound metric is the percentage of total cycles stalled in the frontend due to frontend core resource constraints not related to instruction fetch latency issues caused by memory access components.

Reasons for a stall in the frontend due to the core units are as follows:

- Resteer: Refers to the frontend resteer that cause PC updates, which can either be:
  - A flush that is caused by mispredictions
  - Other machine clears due to microarchitectural reasons such as exceptions and memory hazards
- Flow: Refers to the frontend structural flow stall that is caused in Fetch/Decode unit fetches due to a branch prediction unit that does not provide predictions on time.

This metric breaks down into the following metrics:

- frontend_core_flush_bound
- frontend_core_flow_bound

For further analysis, the next step is the Stage 2 Branch Effectiveness metric group.

Note

Some frontend core bound stall reasons are not captured by the `frontend_core_flush_bound` and `frontend_core_flow_bound` metrics.

3.1.3 Topdown Backend metric group

The Topdown_Backend metric group contains a set of metrics to analyze a backend bound workload.

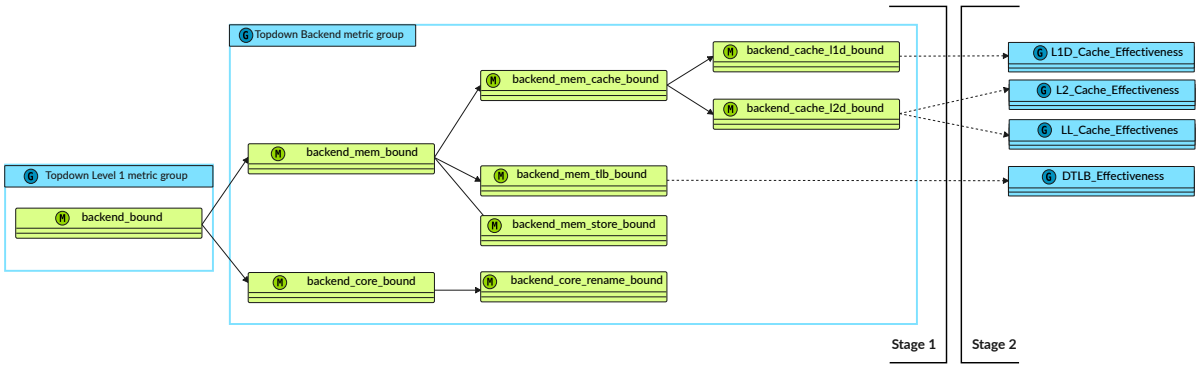
The following implementation criteria apply for this metric group:

- The implementation of the Backend Core Bound metric in Neoverse V3 is the difference between the Backend Bound and Backend Memory Bound metrics. This means that the cycle was stalled in the backend and the stall was not caused by the backend memory units. This derivation is mathematically correct but misses a direct count of the Backend Core Bound metric from an implementation standpoint.
- Due to the implementation, the sum of the Backend Core Bound and Backend Memory Bound metrics is equal to 100% of the Backend Bound metric.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

The following figure shows the metrics for the Topdown_Backend metric group and the next steps in the methodology.

Figure 3-5: Neoverse V3 Metric Group: Topdown_Backend and next steps



The metrics in the Topdown Backend metric group are:

backend_mem_bound

The backend_mem_bound metric is the percentage of total cycles stalled in the backend due to backend core resource constraints related to memory access latency issues caused by memory access components. backend_mem_bound counts the stall cycles in which no operation can be sent to the dispatch unit and a memory operation is being executed.

This metric breaks down into the following metrics:

- backend_mem_cache_bound
  - backend_cache_l1d_bound
  - backend_cache_l2d_bound
- backend_mem_tlb_bound
- backend_mem_store_bound

The following implementation criteria apply for these metrics:

- The Backend_Memory Cache Bound metric is implemented as the number of cycles of either the Backend Memory L1D Bound metric or the Backend Memory L1D Bound metric.
- The Backend Memory L1D Bound metric counts when waiting on an L1 miss but not waiting on an L2 miss.

`backend_mem_*_bound` stall metrics are derived ratios of TLB and Cache stalls such as:

- `backend_mem_l1d_bound` is a ratio of backend memory bound cycles waiting on an L1 data cache miss.
- `backend_mem_l2d_bound` is a ratio of backend memory bound cycles waiting on an L2 unified cache miss.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- `backend_mem_tlb_bound` is a ratio of backend memory bound cycles waiting on data for a TLB miss or walk.

The Backend Memory Store Bound metric also counts backend memory bound stall cycles while stores are outstanding.

For further analysis:

- The next step for `backend_cache_l1d_bound` is the Stage 2 Level 1 Data Cache Effectiveness metric group.
- The next step for `backend_cache_l2d_bound` is the Stage 2 Level 2 Unified Cache Effectiveness metric group and the Stage 2 Last Level Cache Effectiveness metric group.
- The next step for `backend_mem_tlb_bound` is the Stage 2 Data TLB Effectiveness metric group.
- There is no further analysis for for `backend_mem_store_bound`.

backend_core_bound

The backend_core_bound metric is the percentage of total cycles stalled in the backend due to backend core resource constraints not related to instruction fetch latency issues caused by memory access components.

Reasons for a stall in the backend due to the core units are as follows:

- Rename: Refers to rename stalls caused by the register renaming unit that sends operations to the dispatch unit.
- Dispatch Busy: Refers to execution stalls caused by either issue queue unit stalls, execution port starvation, or data dependency issues for operations to make further progress in execution.

For further analysis, the next step is the Stage 1 backend_core_rename_bound metric.

3.2 Stage 2: Microarchitecture exploration

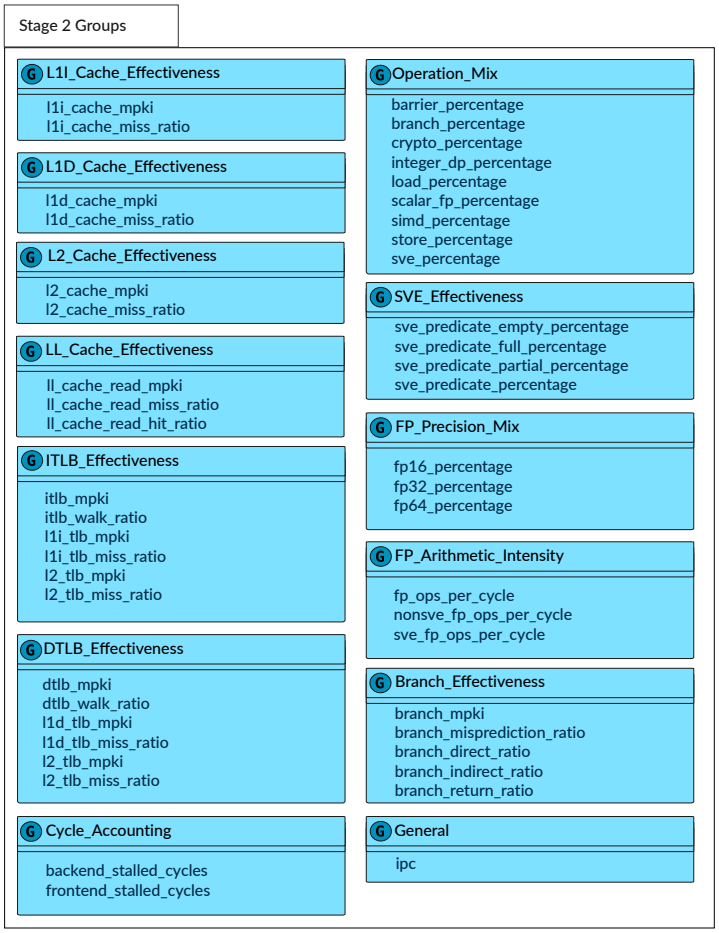
Topdown analysis Stage 2 helps to locate the hot spots in the code and conduct root cause analysis. This stage reveals more details about the CPU component that caused the bottleneck and was identified during the Stage 1 process.

Stage 2 contains metric groups for microarchitecture exploration targeted at CPU resource effectiveness and other relevant metrics. Metrics in this category are designed for users who are interested in the microarchitectural characteristics of key CPU components.

The following figure shows the Stage 2 metric groups for Neoverse V3.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-6: Neoverse V3 Topdown methodology Stage 2 overview



In Neoverse V3 there are two metric groups based on industry standard metrics, Misses Per Kilo Instructions (MPKI) and Miss Ratios, plus eight other Stage 2 metric groups.

Level 1 Data Cache Effectiveness metric group

The L1D_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the L1 Data Cache on the processor.

Level 1 Instruction Cache Effectiveness metric group

The L1I_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the L1 Instruction Cache on the processor.

Level 2 Unified Cache Effectiveness metric group

The L2_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the L2 Unified Cache on the processor.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Last Level Cache Effectiveness metric group

The LL_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the Last Level Cache on the processor.

In systems that support a shared System Level Cache (SLC) in the interconnect that is configured to count Last Level Cache events:

- The Last level cache access, read event counts total SLC accesses made by the core.
- The Last level cache miss, read event counts accesses missed at the SLC.

The ll_cache_read_mpki and ll_cache_read_miss_ratio metrics can be used to analyze the last level read behavior.

Another useful metric to measure the SLC hit percentage for read traffic is ll_cache_read_hit_ratio.

Last level cache events do not have a write variant in Neoverse V3 because the SLC is only used as an eviction cache for the core. In addition, all the writes complete early at the interconnect when the transaction is acknowledged but always completed.

Data TLB Effectiveness and Instruction TLB Effectiveness metric groups

The DTLB_Effectiveness and ITLB_Effectiveness metric groups contain metrics to evaluate the effectiveness of the data TLB and instruction TLB, respectively, on the processor.

An important performance evaluation step is to check the virtual memory system performance, which affects the instruction fetch performance in the frontend and memory access performance on the data side.

The processor translates a virtual address into a physical address for any instruction or data memory access before it accesses the respective cache.


Note

A program's view of memory is the virtual address, but the processor works with the physical address when it accesses cache or memory.

Virtual to physical mappings are defined in the page translation tables which reside in system memory. Access to these tables requires one or more memory operations that take many cycles to complete and are known as a translation table walk. However, TLBs cache these translation table walks, which significantly reduces the number of accesses to system memory and makes translations faster.

Several key metrics can be used to evaluate the TLB effectiveness and the cost of latency that is specifically caused by translation table walks:

- The itlb_mpki and dtlb_mpki metrics provide the rate of TLB Walks per kilo instructions for instruction and data accesses, respectively. These metrics help to evaluate and correlate the TLB efficiency with respect to the total number of instructions.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- The dtlb_walk_ratio metric provides the ratio of DTLB Walks to the overall TLB lookups made by the program. this metric is the same as the DTLB_WALK and MEM_ACCESS events because every MEM_ACCESS causes a L1D_TLB access.
- The itlb_walk_ratio metric provides a percentage of ITLB walks to the overall TLB lookups initiated from the instruction side.

Branch Effectiveness metric group

The Branch_Effectiveness metric group contains metrics to evaluate the effectiveness of branch instruction execution on the processor.

Branch mispredictions are costly in a deeply pipelined CPU, causing pipeline flushes and wasted cycles. Workloads contain, on average, one branch in every six instructions. Although modern CPUs have optimized branch prediction units, there are many use cases that are branch heavy and hard to predict. Examples of such use cases include ray tracing and decision tree algorithms. In some of these cases, there can be hundreds of unique branch paths to take and the target might be input data dependent.

Two key performance metrics can be used for a high-level evaluation of the branch execution performance regarding the overall program execution:

- The branch_mpki metric provides total branch mispredictions per kilo instructions.
- The branch_misprediction_ratio metric can indicate the ratio of branches that were mispredicted to the overall branches.

Branch prediction units work differently depending on the branch type. The main components are:

Branch History Table (BHT)
: Stores the history of conditional branches, taken or not.

Branch Target Buffer (BTB)
: Stores the target address for indirect branches.

Return Address Stack (RAS)
: Stores the function return branches.

Neoverse V3 supports the following three events that categorize the immediate, indirect, and return branches executed, respectively:

- BR_IMMED_SPEC
- BR_INDIRECT_SPEC
- BR_RETURN_SPEC

Getting a breakdown of the branch type helps to investigate each subunit within the branch prediction unit.

Cycle Accounting metric group

The Cycle_Accounting metric group contains a set of metrics that measure the percentage of processor cycles stalled in either frontend or backend of the processor.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

General metric group

The General metric group contains general CPU metrics for performance analysis such as Instructions Per Cycle (IPC).

Floating Point Arithmetic Intensity metric group

The FP_Arithmetic_Intensity metric group provides metrics to evaluate the effectiveness of floating point operation execution on this processor.

The metrics evaluate operations by computation and vector lanes, metrics are provided for scalable vector operations and non scalable vector operations.

Floating Point Precision metric group

The FP_Precision_Mix metric group provides metrics to evaluate tmix of precision of floating point instruction execution on this processor.

SVE Effectiveness metric group

The SVE_Effectiveness metric group provides metrics to evaluate the effectiveness of predicated SVE instruction execution on this processor.

Operation Mix metric group

The Operation_Mix metric group provides the distribution of micro-operation types executed for the program.

The Neoverse V3 microarchitecture has a variety of execution units that can process more than six types of operations:

- Branch
- Single-cycle integers
- Multicycle integers
- Load/store unit with address generation
- Advanced floating-point/SIMD operations
- SVE

The operations that are issued to these execution units can be counted by the PMU events in the Operation Mix metric group.

The following implementation criteria apply for this metric group:

- The `sve_all_percentage` operation includes SVE load and store operations. These operations are also counted in the load_percentage and store_percentage metrics. Therefore, this duplicate counting should be taken into account when these metrics are considered together.

Misses Per Kilo Instructions metric group

The MPKI metric group contains metrics for different CPU resources that can be measured as misses per kilo instructions. These metrics can be used to normalize the misses in CPU components against the total instructions executed. The primary components are branches, caches, and TLBs.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

MPKI is an industry-standard metric that can also help with comparisons across different implementations of the Arm architecture, because instructions retired should count the same on all AArch64-based microarchitectures.


Note

The MPKI metric group is a Stage 2 metric but is not included in the methodology decision tree.

Miss Ratio metric group

The Miss_Ratio metric group contains metrics to measure miss ratios of different processor resources. These metrics can be used to calculate the ratio of misses in CPU components against the total accesses in those components. The primary components are branches, caches, and TLBs.

Miss Ratio metrics provide insights into the efficiency of each CPU component in the pipeline and help to find the root cause of issues.


Note

The Miss_Ratio metric group is a Stage 2 metric but is not included in the methodology decision tree.

3.3 Core memory traffic

The MEM_ACCESS event counts the total number of memory operations that were issued by the Load Store Unit (LSU) of the core.

Because these operations are looked up in the L1D_CACHE first, both the L1D_CACHE and MEM_ACCESS events count at the same rate.

Neoverse V3 also supports two additional events, MEM_ACCESS_RD and MEM_ACCESS_WR that can provide the read and write traffic breakdown respectively. These events are not the same as the LD_SPEC and ST_SPEC events because they count memory operations that are speculatively issued but not always executed.

3.4 Remote cache access

For systems with multiple sockets or SoCs, Neoverse V3 supports the REMOTE_ACCESS event, which counts the memory transactions that were completed by a subordinate source from another chip.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4. Neoverse V3 Telemetry cheat-sheets and lookup tables

The cheat-sheets and lookup tables enable you to find and access metrics and events in different ways.

Cheat-sheets

Both metrics and events are listed by metric groups.

Lookup tables

Metrics are listed alphabetically, with the related events, and metric groups.

Events are listed by code number, with the related metrics, metric groups, and functional groups.

4.1 Metrics cheat sheet for Neoverse V3

Metrics are listed in their respective metric groups. Some metrics are used in more than one metric group.

Neoverse V3 specification provides the following types of metrics:

- Total implemented Common metrics: 67

Topdown Level 1 (4)	Topdown Frontend (8)	Topdown Backend (9)
• backend_bound • bad_speculation • frontend_bound • retiring	• frontend_cache_l1i_bound • frontend_cache_l2i_bound • frontend_core_bound • frontend_core_flow_bound • frontend_core_flush_bound • frontend_mem_bound • frontend_mem_cache_bound • frontend_mem_tlb_bound	• backend_busy_bound • backend_cache_l1d_bound • backend_cache_l2d_bound • backend_core_bound • backend_core_rename_bound • backend_mem_bound • backend_mem_cache_bound • backend_mem_store_bound • backend_mem_tlb_bound

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Cycle Accounting (2)	General (1)	Misses Per Kilo Instructions (10)
• backend_stalled_cycles • frontend_stalled_cycles	• ipc	• branch_mpki • dtlb_mpki • itlb_mpki • l1d_cache_mpki • l1d_tlb_mpki • l1i_cache_mpki • l1i_tlb_mpki • l2_cache_mpki • l2_tlb_mpki • ll_cache_read_mpki

Miss Ratio (10)	SVE Effectiveness (4)	Floating Point Arithmetic Intensity (3)
• branch_misprediction_ratio • dtlb_walk_ratio • itlb_walk_ratio • l1d_cache_miss_ratio • l1d_tlb_miss_ratio • l1i_cache_miss_ratio • l1i_tlb_miss_ratio • l2_cache_miss_ratio • l2_tlb_miss_ratio • ll_cache_read_miss_ratio	• sve_predicate_empty_percentage • sve_predicate_full_percentage • sve_predicate_partial_percentage • sve_predicate_percentage	• fp_ops_per_cycle • nonsve_fp_ops_per_cycle • sve_fp_ops_per_cycle

Floating Point Precision (3)	Branch Effectiveness (5)	Instruction TLB Effectiveness (6)
• fp16_percentage • fp32_percentage • fp64_percentage	• branch_direct_ratio • branch_indirect_ratio • branch_misprediction_ratio • branch_mpki • branch_return_ratio	• itlb_mpki • itlb_walk_ratio • l1i_tlb_miss_ratio • l1i_tlb_mpki • l2_tlb_miss_ratio • l2_tlb_mpki

Data TLB Effectiveness (6)	L1 Instruction Cache Effectiveness (2)	L1 Data Cache Effectiveness (2)
• dtlb_mpki • dtlb_walk_ratio • l1d_tlb_miss_ratio • l1d_tlb_mpki • l2_tlb_miss_ratio • l2_tlb_mpki	• l1i_cache_miss_ratio • l1i_cache_mpki	• l1d_cache_miss_ratio • l1d_cache_mpki

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

L2 Unified Cache Effectiveness (2)	Last Level Cache Effectiveness (3)	Speculative Operation Mix (9)
• l2_cache_miss_ratio • l2_cache_mpki	• ll_cache_read_hit_ratio • ll_cache_read_miss_ratio • ll_cache_read_mpki	• barrier_percentage • branch_percentage • crypto_percentage • integer_dp_percentage • load_percentage • scalar_fp_percentage • simd_percentage • store_percentage • sve_all_percentage

4.2 PMU events cheat sheet for Neoverse V3

Events are listed in their respective metric groups. Some events are not used in the Methodology, therefore are not shown in the cheat sheet.

Neoverse V3 specification provides the following types of PMU events:

- Total implemented Common events: 226
- Total Implemented Product ImpDef events: 27
- PMU Only events : 27
- ETE Only events : 0

Topdown Level 1 (7)	Topdown Frontend (8)	Topdown Backend (9)
• CPU_CYCLES • OP_RETIRED • OP_SPEC • STALL_FRONTEND_FLUSH • STALL_SLOT • STALL_SLOT_BACKEND • STALL_SLOT_FRONTEND	• STALL_FRONTEND • STALL_FRONTEND_CPUBOUND • STALL_FRONTEND_FLOW • STALL_FRONTEND_FLUSH • STALL_FRONTEND_L1I • STALL_FRONTEND_MEM • STALL_FRONTEND_MEMBOUND • STALL_FRONTEND_TLB	• STALL_BACKEND • STALL_BACKEND_BUSY • STALL_BACKEND_CPUBOUND • STALL_BACKEND_L1D • STALL_BACKEND_MEM • STALL_BACKEND_MEMBOUND • STALL_BACKEND_RENAME • STALL_BACKEND_ST • STALL_BACKEND_TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Cycle Accounting (3)	General (2)	Misses Per Kilo Instructions (11)
• CPU_CYCLES • STALL_BACKEND • STALL_FRONTEND	• CPU_CYCLES • INST_RETIRED	• BR_MIS_PRED_RETIRED • DTLB_WALK • INST_RETIRED • ITLB_WALK • L1D_CACHE_REFILL • L1D_TLB_REFILL • L1I_CACHE_REFILL • L1I_TLB_REFILL • L2D_CACHE_REFILL • L2D_TLB_REFILL • LL_CACHE_MISS_RD

Miss Ratio (18)	SVE Effectiveness (5)	Floating Point Arithmetic Intensity (3)
• BR_MIS_PRED_RETIRED • BR_RETIRED • DTLB_WALK • ITLB_WALK • L1D_CACHE • L1D_CACHE_REFILL • L1D_TLB • L1D_TLB_REFILL • L1I_CACHE • L1I_CACHE_REFILL • L1I_TLB • L1I_TLB_REFILL • L2D_CACHE • L2D_CACHE_REFILL • L2D_TLB • L2D_TLB_REFILL • LL_CACHE_MISS_RD • LL_CACHE_RD	• INST_SPEC • SVE_PRED_EMPTY_SPEC • SVE_PRED_FULL_SPEC • SVE_PRED_PARTIAL_SPEC • SVE_PRED_SPEC	• CPU_CYCLES • FP_FIXED_OPS_SPEC • FP_SCALE_OPS_SPEC

Floating Point Precision (4)	Branch Effectiveness (6)	Instruction TLB Effectiveness (6)
• FP_DP_SPEC • FP_HP_SPEC • FP_SP_SPEC • INST_SPEC	• BR_IMMED_RETIRED • BR_IND_RETIRED • BR_MIS_PRED_RETIRED • BR_RETIRED • BR_RETURN_RETIRED • INST_RETIRED	• INST_RETIRED • ITLB_WALK • L1I_TLB • L1I_TLB_REFILL • L2D_TLB • L2D_TLB_REFILL

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Data TLB Effectiveness (6)	L1 Instruction Cache Effectiveness (3)	L1 Data Cache Effectiveness (3)
• DTLB_WALK • INST_RETIRED • L1D_TLB • L1D_TLB_REFILL • L2D_TLB • L2D_TLB_REFILL	• INST_RETIRED • L1I_CACHE • L1I_CACHE_REFILL	• INST_RETIRED • L1D_CACHE • L1D_CACHE_REFILL

L2 Unified Cache Effectiveness (3)	Last Level Cache Effectiveness (3)	Speculative Operation Mix (13)
• INST_RETIRED • L2D_CACHE • L2D_CACHE_REFILL	• INST_RETIRED • LL_CACHE_MISS_RD • LL_CACHE_RD	• ASE_SPEC • BR_IMMED_SPEC • BR_INDIRECT_SPEC • CRYPTO_SPEC • DMB_SPEC • DP_SPEC • DSB_SPEC • INST_SPEC • ISB_SPEC • LD_SPEC • ST_SPEC • SVE_INST_SPEC • VFP_SPEC

4.3 Metrics lookup table for Neoverse V3

All metrics are listed alphabetically, with the related events, and metric groups. Some metrics are used in more than one metric group, in that case they are listed multiple times so that you can jump to the most relevant metric group for your requirements.

Table 4-13: Metrics listed by name, with related events and metric groups

Metric Name	Formula from Events	Metric Groups
backend_bound	STALL_SLOT_BACKEND / (10 * CPU_CYCLES) * 100	• Topdown_L1
backend_busy_bound	STALL_BACKEND_BUSY / STALL_BACKEND * 100	• Topdown_Backend
backend_cache_l1d_bound	STALL_BACKEND_L1D / (STALL_BACKEND_L1D + STALL_BACKEND_MEM) * 100	• Topdown_Backend
backend_cache_l2d_bound	STALL_BACKEND_MEM / (STALL_BACKEND_L1D + STALL_BACKEND_MEM) * 100	• Topdown_Backend
backend_core_bound	STALL_BACKEND_CPUBOUND / STALL_BACKEND * 100	• Topdown_Backend

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric Name	Formula from Events	Metric Groups
backend_core_rename_bound	STALL_BACKEND_RENAME / STALL_BACKEND_CPUBOUND * 100	• Topdown_Backend
backend_mem_bound	STALL_BACKEND_MEMBOUND / STALL_BACKEND * 100	• Topdown_Backend
backend_mem_cache_bound	(STALL_BACKEND_L1D + STALL_BACKEND_MEM) / STALL_BACKEND_MEMBOUND * 100	• Topdown_Backend
backend_mem_store_bound	STALL_BACKEND_ST / STALL_BACKEND_MEMBOUND * 100	• Topdown_Backend
backend_mem_tlb_bound	STALL_BACKEND_TLB / STALL_BACKEND_MEMBOUND * 100	• Topdown_Backend
backend_stalled_cycles	STALL_BACKEND / CPU_CYCLES * 100	• Cycle_Accounting
bad_speculation	(1 - STALL_SLOT / (10 * CPU_CYCLES)) * (1 - OP_RETIRED / OP_SPEC) * 100 + STALL_FRONTEND_FLUSH / CPU_CYCLES * 100	• Topdown_L1
barrier_percentage	(ISB_SPEC + DSB_SPEC + DMB_SPEC) / INST_SPEC * 100	• Operation_Mix
branch_direct_ratio	BR_IMMED_RETIRED / BR_RETIRED	• Branch_Effectiveness
branch_indirect_ratio	BR_IND_RETIRED / BR_RETIRED	• Branch_Effectiveness
• branch_misprediction_ratio in Branch_Effectiveness • branch_misprediction_ratio in Miss_Ratio	BR_MIS_PRED_RETIRED / BR_RETIRED	• Branch_Effectiveness • Miss_Ratio
• branch_mpki in Branch_Effectiveness • branch_mpki in MPKI	BR_MIS_PRED_RETIRED / INST_RETIRED * 1000	• Branch_Effectiveness • MPKI
branch_percentage	(BR_IMMED_SPEC + BR_INDIRECT_SPEC) / INST_SPEC * 100	• Operation_Mix
branch_return_ratio	BR_RETURN_RETIRED / BR_RETIRED	• Branch_Effectiveness
crypto_percentage	CRYPTO_SPEC / INST_SPEC * 100	• Operation_Mix
• dtlb_mpki in DTLB_Effectiveness • dtlb_mpki in MPKI	DTLB_WALK / INST_RETIRED * 1000	• DTLB_Effectiveness • MPKI
• dtlb_walk_ratio in DTLB_Effectiveness • dtlb_walk_ratio in Miss_Ratio	DTLB_WALK / L1D_TLB	• DTLB_Effectiveness • Miss_Ratio
fp16_percentage	FP_HP_SPEC / INST_SPEC * 100	• FP_Precision_Mix
fp32_percentage	FP_SP_SPEC / INST_SPEC * 100	• FP_Precision_Mix
fp64_percentage	FP_DP_SPEC / INST_SPEC * 100	• FP_Precision_Mix
fp_ops_per_cycle	(FP_SCALE_OPS_SPEC + FP_FIXED_OPS_SPEC) / CPU_CYCLES	• FP_Arithmetic_Intensity
frontend_bound	(STALL_SLOT_FRONTEND / (10 * CPU_CYCLES) - STALL_FRONTEND_FLUSH / CPU_CYCLES) * 100	• Topdown_L1
frontend_cache_l1i_bound	STALL_FRONTEND_L1I / (STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) * 100	• Topdown_Frontend

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric Name	Formula from Events	Metric Groups
frontend_cache_l2i_bound	STALL_FRONTEND_MEM / (STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) * 100	• Topdown_Frontend
frontend_core_bound	STALL_FRONTEND_CPUBOUND / STALL_FRONTEND * 100	• Topdown_Frontend
frontend_core_flow_bound	STALL_FRONTEND_FLOW / STALL_FRONTEND_CPUBOUND * 100	• Topdown_Frontend
frontend_core_flush_bound	STALL_FRONTEND_FLUSH / STALL_FRONTEND_CPUBOUND * 100	• Topdown_Frontend
frontend_mem_bound	STALL_FRONTEND_MEMBOUND / STALL_FRONTEND * 100	• Topdown_Frontend
frontend_mem_cache_bound	(STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) / STALL_FRONTEND_MEMBOUND * 100	• Topdown_Frontend
frontend_mem_tlb_bound	STALL_FRONTEND_TLB / STALL_FRONTEND_MEMBOUND * 100	• Topdown_Frontend
frontend_stalled_cycles	STALL_FRONTEND / CPU_CYCLES * 100	• Cycle_Accounting
integer_dp_percentage	DP_SPEC / INST_SPEC * 100	• Operation_Mix
ipc	INST_RETIRED / CPU_CYCLES	• General
• itlb_mpki in ITLB_Effectiveness • itlb_mpki in MPKI	ITLB_WALK / INST_RETIRED * 1000	• ITLB_Effectiveness • MPKI
• itlb_walk_ratio in ITLB_Effectiveness • itlb_walk_ratio in Miss_Ratio	ITLB_WALK / L1I_TLB	• ITLB_Effectiveness • Miss_Ratio
• l1d_cache_miss_ratio in L1D_Cache_Effectiveness • l1d_cache_miss_ratio in Miss_Ratio	L1D_CACHE_REFILL / L1D_CACHE	• L1D_Cache_Effectiveness • Miss_Ratio
• l1d_cache_mpki in L1D_Cache_Effectiveness • l1d_cache_mpki in MPKI	L1D_CACHE_REFILL / INST_RETIRED * 1000	• L1D_Cache_Effectiveness • MPKI
• l1d_tlb_miss_ratio in DTLB_Effectiveness • l1d_tlb_miss_ratio in Miss_Ratio	L1D_TLB_REFILL / L1D_TLB	• DTLB_Effectiveness • Miss_Ratio
• l1d_tlb_mpki in DTLB_Effectiveness • l1d_tlb_mpki in MPKI	L1D_TLB_REFILL / INST_RETIRED * 1000	• DTLB_Effectiveness • MPKI
• l1i_cache_miss_ratio in L1I_Cache_Effectiveness • l1i_cache_miss_ratio in Miss_Ratio	L1I_CACHE_REFILL / L1I_CACHE	• L1I_Cache_Effectiveness • Miss_Ratio
• l1i_cache_mpki in L1I_Cache_Effectiveness • l1i_cache_mpki in MPKI	L1I_CACHE_REFILL / INST_RETIRED * 1000	• L1I_Cache_Effectiveness • MPKI
• l1i_tlb_miss_ratio in ITLB_Effectiveness • l1i_tlb_miss_ratio in Miss_Ratio	L1I_TLB_REFILL / L1I_TLB	• ITLB_Effectiveness • Miss_Ratio
• l1i_tlb_mpki in ITLB_Effectiveness • l1i_tlb_mpki in MPKI	L1I_TLB_REFILL / INST_RETIRED * 1000	• ITLB_Effectiveness • MPKI

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric Name	Formula from Events	Metric Groups
• l2_cache_miss_ratio in L2_Cache_Effectiveness • l2_cache_miss_ratio in Miss_Ratio	L2D_CACHE_REFILL / L2D_CACHE	• L2_Cache_Effectiveness • Miss_Ratio
• l2_cache_mpki in L2_Cache_Effectiveness • l2_cache_mpki in MPKI	L2D_CACHE_REFILL / INST_RETIRED * 1000	• L2_Cache_Effectiveness • MPKI
• l2_tlb_miss_ratio in DTLB_Effectiveness • l2_tlb_miss_ratio in ITLB_Effectiveness • l2_tlb_miss_ratio in Miss_Ratio	L2D_TLB_REFILL / L2D_TLB	• DTLB_Effectiveness • ITLB_Effectiveness • Miss_Ratio
• l2_tlb_mpki in DTLB_Effectiveness • l2_tlb_mpki in ITLB_Effectiveness • l2_tlb_mpki in MPKI	L2D_TLB_REFILL / INST_RETIRED * 1000	• DTLB_Effectiveness • ITLB_Effectiveness • MPKI
ll_cache_read_hit_ratio	(LL_CACHE_RD - LL_CACHE_MISS_RD) / LL_CACHE_RD	• LL_Cache_Effectiveness
• ll_cache_read_miss_ratio in LL_Cache_Effectiveness • ll_cache_read_miss_ratio in Miss_Ratio	LL_CACHE_MISS_RD / LL_CACHE_RD	• LL_Cache_Effectiveness • Miss_Ratio
• ll_cache_read_mpki in LL_Cache_Effectiveness • ll_cache_read_mpki in MPKI	LL_CACHE_MISS_RD / INST_RETIRED * 1000	• LL_Cache_Effectiveness • MPKI
load_percentage	LD_SPEC / INST_SPEC * 100	• Operation_Mix
nonsve_fp_ops_per_cycle	FP_FIXED_OPS_SPEC / CPU_CYCLES	• FP_Arithmetic_Intensity
retiring	(1 - STALL_SLOT / (CPU_CYCLES * 10)) * (OP_RETIRED / OP_SPEC) * 100	• Topdown_L1
scalar_fp_percentage	VFP_SPEC / INST_SPEC * 100	• Operation_Mix
simd_percentage	ASE_SPEC / INST_SPEC * 100	• Operation_Mix
store_percentage	ST_SPEC / INST_SPEC * 100	• Operation_Mix
sve_all_percentage	SVE_INST_SPEC / INST_SPEC * 100	• Operation_Mix
sve_fp_ops_per_cycle	FP_SCALE_OPS_SPEC / CPU_CYCLES	• FP_Arithmetic_Intensity
sve_predicate_empty_percentage	SVE_PRED_EMPTY_SPEC / SVE_PRED_SPEC * 100	• SVE_Effectiveness
sve_predicate_full_percentage	SVE_PRED_FULL_SPEC / SVE_PRED_SPEC * 100	• SVE_Effectiveness
sve_predicate_partial_percentage	SVE_PRED_PARTIAL_SPEC / SVE_PRED_SPEC * 100	• SVE_Effectiveness
sve_predicate_percentage	SVE_PRED_SPEC / INST_SPEC * 100	• SVE_Effectiveness

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.4 PMU events lookup table for Neoverse V3

All events are listed in event code order, with the related metrics, metric groups, and functional groups. Some events are not used in the Methodology, however, they are all listed for completeness.

Summary of Events:

- Total Possible Common events: 761
- Total implemented Common events: 226
  - Common : Architectural-defined events: 162
  - Common : Implementation-defined events: 64
- Total Implemented Product ImpDef events: 27
- PMU Only Events : 27
- ETE Only Events : 0

Table 4-14: Events listed by Event Code, with related Metrics, Metric Groups, and Functional Groups

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0000, SW_INCR	-	-	• Retired
0x0001, L1I_CACHE_REFILL	• l1i_cache_mpki in L1I_Cache _Effectiveness • l1i_cache_mpki in MPKI • l1i_cache_miss_ratio in L1I_Cache _Effectiveness • l1i_cache_miss_ratio in Miss_Ratio	• L1I_Cache _Effectiveness • MPKI • Miss_Ratio	• L1I_Cache
0x0002, L1I_TLB_REFILL	• l1i_tlb_mpki in ITLB_Effectiveness • l1i_tlb_mpki in MPKI • l1i_tlb_miss_ratio in ITLB_Effectiveness • l1i_tlb_miss_ratio in Miss_Ratio	• ITLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x0003, L1D_CACHE_REFILL	• l1d_cache_mpki in L1D_Cache _Effectiveness • l1d_cache_mpki in MPKI • l1d_cache_miss_ratio in L1D_Cache _Effectiveness • l1d_cache_miss_ratio in Miss_Ratio	• L1D_Cache _Effectiveness • MPKI • Miss_Ratio	• L1D_Cache
0x0004, L1D_CACHE	• l1d_cache_miss_ratio in L1D_Cache _Effectiveness • l1d_cache_miss_ratio in Miss_Ratio	• L1D_Cache _Effectiveness • Miss_Ratio	• L1D_Cache

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0005, L1D_TLB_REFILL	• l1d_tlb_mpki in DTLB_Effectiveness • l1d_tlb_mpki in MPKI • l1d_tlb_miss_ratio in DTLB_Effectiveness • l1d_tlb_miss_ratio in Miss_Ratio	• DTLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x0008, INST_RETIRED	• ipc • branch_mpki in Branch_Effectiveness • branch_mpki in MPKI • itlb_mpki in ITLB_Effectiveness • itlb_mpki in MPKI • l1i_tlb_mpki in ITLB_Effectiveness • l1i_tlb_mpki in MPKI • dtlb_mpki in DTLB_Effectiveness • dtlb_mpki in MPKI • l1d_tlb_mpki in DTLB_Effectiveness • l1d_tlb_mpki in MPKI • l2_tlb_mpki in DTLB_Effectiveness • l2_tlb_mpki in ITLB_Effectiveness • l2_tlb_mpki in MPKI • l1i_cache_mpki in L1I_Cache _Effectiveness • l1i_cache_mpki in MPKI • l1d_cache_mpki in L1D_Cache _Effectiveness • l1d_cache_mpki in MPKI • l2_cache_mpki in L2_Cache _Effectiveness • l2_cache_mpki in MPKI • ll_cache_read_mpki in LL_Cache _Effectiveness • ll_cache_read_mpki in MPKI	• Branch_Effectiveness • DTLB_Effectiveness • General • ITLB_Effectiveness • L1D_Cache _Effectiveness • L1I_Cache _Effectiveness • L2_Cache _Effectiveness • LL_Cache _Effectiveness • MPKI	• Retired
0x0009, EXC_TAKEN	-	-	• Exception
0x000A, EXC_RETURN	-	-	• Exception
0x000B, CID_WRITE_RETIRED	-	-	• Retired
0x000D, BR_IMMED_RETIRED	• branch_direct_ratio	• Branch_Effectiveness	• Retired
0x000E, BR_RETURN_RETIRED	• branch_return_ratio	• Branch_Effectiveness	• Retired

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0010, BR_MIS_PRED	-	-	• Spec_Operation
0x0011, CPU_CYCLES	• frontend_stalled_cycles • backend_stalled_cycles • frontend_bound • backend_bound • retiring • bad_speculation • ipc • sve_fp_ops_per_cycle • nonsve_fp_ops_per_cycle • fp_ops_per_cycle	• Cycle_Accounting • FP_Arithmetic _Intensity • General • Topdown_L1	• General
0x0012, BR_PRED	-	-	• Spec_Operation
0x0013, MEM_ACCESS	-	-	• Memory
0x0014, L1I_CACHE	• l1i_cache_miss_ratio in L1I_Cache _Effectiveness • l1i_cache_miss_ratio in Miss_Ratio	• L1I_Cache _Effectiveness • Miss_Ratio	• L1I_Cache
0x0015, L1D_CACHE_WB	-	-	• L1D_Cache
0x0016, L2D_CACHE	• l2_cache_miss_ratio in L2_Cache _Effectiveness • l2_cache_miss_ratio in Miss_Ratio	• L2_Cache _Effectiveness • Miss_Ratio	• L2_Cache
0x0017, L2D_CACHE_REFILL	• l2_cache_mpki in L2_Cache _Effectiveness • l2_cache_mpki in MPKI • l2_cache_miss_ratio in L2_Cache _Effectiveness • l2_cache_miss_ratio in Miss_Ratio	• L2_Cache _Effectiveness • MPKI • Miss_Ratio	• L2_Cache
0x0018, L2D_CACHE_WB	-	-	• L2_Cache
0x0019, BUS_ACCESS	-	-	• Bus
0x001A, MEMORY_ERROR	-	-	• Memory

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x001B, INST_SPEC	• load_percentage • store_percentage • integer_dp_percentage • simd_percentage • scalar_fp_percentage • barrier_percentage • branch_percentage • crypto_percentage • sve_all_percentage • sve_predicate_percentage • fp16_percentage • fp32_percentage • fp64_percentage	• FP_Precision_Mix • Operation_Mix • SVE_Effectiveness	• Spec_Operation
0x001C, TTBR_WRITE_RETIRED	-	-	• Retired
0x001D, BUS_CYCLES	-	-	• Bus
0x001E, CHAIN	-	-	• Chain
0x0021, BR_RETIRED	• branch_direct_ratio • branch_indirect_ratio • branch_return_ratio • branch_misprediction _ratio in Branch_Effectiveness • branch_misprediction _ratio in Miss_Ratio	• Branch_Effectiveness • Miss_Ratio	• Retired
0x0022, BR_MIS_PRED_RETIRED	• branch_mpki in Branch_Effectiveness • branch_mpki in MPKI • branch_misprediction _ratio in Branch_Effectiveness • branch_misprediction _ratio in Miss_Ratio	• Branch_Effectiveness • MPKI • Miss_Ratio	• Retired
0x0023, STALL_FRONTEND	• frontend_stalled_cycles • frontend_core_bound • frontend_mem_bound	• Cycle_Accounting • Topdown_Frontend	• Stall
0x0024, STALL_BACKEND	• backend_stalled_cycles • backend_core_bound • backend_mem_bound • backend_busy_bound	• Cycle_Accounting • Topdown_Backend	• Stall

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0025, L1D_TLB	• dtlb_walk_ratio in DTLB_Effectiveness • dtlb_walk_ratio in Miss_Ratio • l1d_tlb_miss_ratio in DTLB_Effectiveness • l1d_tlb_miss_ratio in Miss_Ratio	• DTLB_Effectiveness • Miss_Ratio	• TLB
0x0026, L1I_TLB	• itlb_walk_ratio in ITLB_Effectiveness • itlb_walk_ratio in Miss_Ratio • l1i_tlb_miss_ratio in ITLB_Effectiveness • l1i_tlb_miss_ratio in Miss_Ratio	• ITLB_Effectiveness • Miss_Ratio	• TLB
0x002D, L2D_TLB_REFILL	• l2_tlb_mpki in DTLB_Effectiveness • l2_tlb_mpki in ITLB_Effectiveness • l2_tlb_mpki in MPKI • l2_tlb_miss_ratio in DTLB_Effectiveness • l2_tlb_miss_ratio in ITLB_Effectiveness • l2_tlb_miss_ratio in Miss_Ratio	• DTLB_Effectiveness • ITLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x002F, L2D_TLB	• l2_tlb_miss_ratio in DTLB_Effectiveness • l2_tlb_miss_ratio in ITLB_Effectiveness • l2_tlb_miss_ratio in Miss_Ratio	• DTLB_Effectiveness • ITLB_Effectiveness • Miss_Ratio	• TLB
0x0031, REMOTE_ACCESS	-	-	• Memory
0x0034, DTLB_WALK	• dtlb_mpki in DTLB_Effectiveness • dtlb_mpki in MPKI • dtlb_walk_ratio in DTLB_Effectiveness • dtlb_walk_ratio in Miss_Ratio	• DTLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x0035, ITLB_WALK	• itlb_mpki in ITLB_Effectiveness • itlb_mpki in MPKI • itlb_walk_ratio in ITLB_Effectiveness • itlb_walk_ratio in Miss_Ratio	• ITLB_Effectiveness • MPKI • Miss_Ratio	• TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0036, LL_CACHE_RD	• ll_cache_read_miss_ratio in LL_Cache _Effectiveness • ll_cache_read_miss_ratio in Miss_Ratio • ll_cache_read_hit_ratio	• LL_Cache _Effectiveness • Miss_Ratio	• LL_Cache
0x0037, LL_CACHE_MISS_RD	• ll_cache_read_mpki in LL_Cache _Effectiveness • ll_cache_read_mpki in MPKI • ll_cache_read_miss_ratio in LL_Cache _Effectiveness • ll_cache_read_miss_ratio in Miss_Ratio • ll_cache_read_hit_ratio	• LL_Cache _Effectiveness • MPKI • Miss_Ratio	• LL_Cache
0x0039, L1D_CACHE_LMISS_RD	-	-	• L1D_Cache
0x003A, OP_RETIRED	• retiring • bad_speculation	• Topdown_L1	• Retired
0x003B, OP_SPEC	• retiring • bad_speculation	• Topdown_L1	• Spec_Operation
0x003C, STALL	-	-	• Stall
0x003D, STALL_SLOT_BACKEND	• backend_bound	• Topdown_L1	• Stall
0x003E, STALL_SLOT_FRONTEND	• frontend_bound	• Topdown_L1	• Stall
0x003F, STALL_SLOT	• retiring • bad_speculation	• Topdown_L1	• Stall
0x0040, L1D_CACHE_RD	-	-	• L1D_Cache
0x0041, L1D_CACHE_WR	-	-	• L1D_Cache
0x0042, L1D_CACHE_REFILL_RD	-	-	• L1D_Cache
0x0043, L1D_CACHE_REFILL_WR	-	-	• L1D_Cache
0x0044, L1D_CACHE_REFILL_INNER	-	-	• L1D_Cache
0x0045, L1D_CACHE_REFILL_OUTER	-	-	• L1D_Cache
0x0046, L1D_CACHE_WB_VICTIM	-	-	• L1D_Cache
0x0047, L1D_CACHE_WB_CLEAN	-	-	• L1D_Cache
0x0048, L1D_CACHE_INVAL	-	-	• L1D_Cache
0x004C, L1D_TLB_REFILL_RD	-	-	• TLB
0x004D, L1D_TLB_REFILL_WR	-	-	• TLB
0x004E, L1D_TLB_RD	-	-	• TLB
0x004F, L1D_TLB_WR	-	-	• TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0050, L2D_CACHE_RD	-	-	• L2_Cache
0x0051, L2D_CACHE_WR	-	-	• L2_Cache
0x0052, L2D_CACHE_REFILL_RD	-	-	• L2_Cache
0x0053, L2D_CACHE_REFILL_WR	-	-	• L2_Cache
0x0056, L2D_CACHE_WB_VICTIM	-	-	• L2_Cache
0x0057, L2D_CACHE_WB_CLEAN	-	-	• L2_Cache
0x0058, L2D_CACHE_INVAL	-	-	• L2_Cache
0x005C, L2D_TLB_REFILL_RD	-	-	• TLB
0x005D, L2D_TLB_REFILL_WR	-	-	• TLB
0x005E, L2D_TLB_RD	-	-	• TLB
0x005F, L2D_TLB_WR	-	-	• TLB
0x0060, BUS_ACCESS_RD	-	-	• Bus
0x0061, BUS_ACCESS_WR	-	-	• Bus
0x0066, MEM_ACCESS_RD	-	-	• Memory
0x0067, MEM_ACCESS_WR	-	-	• Memory
0x0068, UNALIGNED_LD_SPEC	-	-	• Spec_Operation
0x0069, UNALIGNED_ST_SPEC	-	-	• Spec_Operation
0x006A, UNALIGNED_LDST_SPEC	-	-	• Spec_Operation
0x006C, LDREX_SPEC	-	-	• Spec_Operation
0x006D, STREX_PASS_SPEC	-	-	• Spec_Operation
0x006E, STREX_FAIL_SPEC	-	-	• Spec_Operation
0x006F, STREX_SPEC	-	-	• Spec_Operation
0x0070, LD_SPEC	• load_percentage	• Operation_Mix	• Spec_Operation
0x0071, ST_SPEC	• store_percentage	• Operation_Mix	• Spec_Operation
0x0072, LDST_SPEC	-	-	• Spec_Operation
0x0073, DP_SPEC	• integer_dp_percentage	• Operation_Mix	• Spec_Operation
0x0074, ASE_SPEC	• simd_percentage	• Operation_Mix	• Spec_Operation
0x0075, VFP_SPEC	• scalar_fp_percentage	• Operation_Mix	• Spec_Operation
0x0076, PC_WRITE_SPEC	-	-	• Spec_Operation
0x0077, CRYPTO_SPEC	• crypto_percentage	• Operation_Mix	• Spec_Operation
0x0078, BR_IMMED_SPEC	• branch_percentage	• Operation_Mix	• Spec_Operation
0x0079, BR_RETURN_SPEC	-	-	• Spec_Operation
0x007A, BR_INDIRECT_SPEC	• branch_percentage	• Operation_Mix	• Spec_Operation
0x007C, ISB_SPEC	• barrier_percentage	• Operation_Mix	• Spec_Operation
0x007D, DSB_SPEC	• barrier_percentage	• Operation_Mix	• Spec_Operation
0x007E, DMB_SPEC	• barrier_percentage	• Operation_Mix	• Spec_Operation
0x0081, EXC_UNDEF	-	-	• Exception

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0082, EXC_SVC	-	-	• Exception
0x0083, EXC_PABORT	-	-	• Exception
0x0084, EXC_DABORT	-	-	• Exception
0x0086, EXC_IRQ	-	-	• Exception
0x0087, EXC_FIQ	-	-	• Exception
0x0088, EXC_SMC	-	-	• Exception
0x008A, EXC_HVC	-	-	• Exception
0x008B, EXC_TRAP_PABORT	-	-	• Exception
0x008C, EXC_TRAP_DABORT	-	-	• Exception
0x008D, EXC_TRAP_OTHER	-	-	• Exception
0x008E, EXC_TRAP_IRQ	-	-	• Exception
0x008F, EXC_TRAP_FIQ	-	-	• Exception
0x0090, RC_LD_SPEC	-	-	• Spec_Operation
0x0091, RC_ST_SPEC	-	-	• Spec_Operation
0x0108, IMP_L2_CACHE_REFILL_IF	-	-	• CPU_Debug
0x010B, IMP_L2_CACHE_PREFETCH_LATE	-	-	• CPU_Debug
0x0120, IMP_CT_FLUSH	-	-	• CPU_Debug
0x0121, IMP_CT_FLUSH_MEM_HAZARD	-	-	• CPU_Debug
0x0122, IMP_CT_FLUSH_BAD_BRANCH	-	-	• CPU_Debug
0x0123, IMP_CT_FLUSH_PREDECODE_ERR	-	-	• CPU_Debug
0x0124, IMP_CT_FLUSH_ISB	-	-	• CPU_Debug
0x0125, IMP_CT_FLUSH_OTHER	-	-	• CPU_Debug
0x0127, IMP_LS_RAR_HAZARD	-	-	• CPU_Debug
0x0128, IMP_LS_RAW_HAZARD	-	-	• CPU_Debug
0x0158, IMP_STALL_BACKEND_RENAME_FRF	-	-	• CPU_Debug
0x0159, IMP_STALL_BACKEND_RENAME_GRF	-	-	• CPU_Debug
0x015A, IMP_STALL_BACKEND_RENAME_VRF	-	-	• CPU_Debug
0x015C, IMP_STALL_BACKEND_IQ _SX	-	-	• Stall
0x015D, IMP_STALL_BACKEND_IQ _MX	-	-	• Stall
0x015E, IMP_STALL_BACKEND_IQ _LS	-	-	• Stall

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x015F, IMP_STALL_BACKEND_IQ _VX	-	-	• Stall
0x0160, IMP_STALL_BACKEND_MCQ	-	-	• Stall
0x017B, IMP_NEAR_CAS	-	-	• CPU_Debug
0x017C, IMP_NEAR_CAS_PASS	-	-	• CPU_Debug
0x0198, IMP_L2_CHI_RX_CBUSY_0	-	-	• General
0x0199, IMP_L2_CHI_RX_CBUSY_1	-	-	• General
0x019A, IMP_L2_CHI_RX_CBUSY_2	-	-	• General
0x019B, IMP_L2_CHI_RX_CBUSY_3	-	-	• General
0x019C, IMP_L2_CHI_RX_CBUSY_MT	-	-	• General
0x01B8, IMP_L2D_CACHE_L1HWPRF	-	-	• L2_Cache
0x01B9, IMP_L2D_CACHE_REFILL _L1HWPRF	-	-	• L2_Cache
0x4000, SAMPLE_POP	-	-	• SPE
0x4001, SAMPLE_FEED	-	-	• SPE
0x4002, SAMPLE_FILTRATE	-	-	• SPE
0x4003, SAMPLE_COLLISION	-	-	• SPE
0x4004, CNT_CYCLES	-	-	• General
0x4005, STALL_BACKEND_MEM	• backend_mem_cache_bound • backend_cache_l1d_bound • backend_cache_l2d_bound	• Topdown_Backend	• Stall
0x4006, L1I_CACHE_LMISS	-	-	• L1I_Cache
0x4009, L2D_CACHE_LMISS_RD	-	-	• L2_Cache
0x4020, LDST_ALIGN_LAT	-	-	• Memory
0x4021, LD_ALIGN_LAT	-	-	• Memory
0x4022, ST_ALIGN_LAT	-	-	• Memory
0x4024, MEM_ACCESS_CHECKED	-	-	• Memory
0x4025, MEM_ACCESS_CHECKED_RD	-	-	• Memory
0x4026, MEM_ACCESS_CHECKED_WR	-	-	• Memory
0x8004, SIMD_INST_SPEC	-	-	• Spec_Operation
0x8005, ASE_INST_SPEC	-	-	• Spec_Operation
0x8006, SVE_INST_SPEC	• sve_all_percentage	• Operation_Mix	• SVE

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8014, FP_HP_SPEC	• fp16_percentage	• FP_Precision_Mix	• FP_Operation
0x8018, FP_SP_SPEC	• fp32_percentage	• FP_Precision_Mix	• FP_Operation
0x801C, FP_DP_SPEC	• fp64_percentage	• FP_Precision_Mix	• FP_Operation
0x8040, INT_SPEC	-	-	• Spec_Operation
0x8074, SVE_PRED_SPEC	• sve_predicate_percentage • sve_predicate_full _percentage • sve_predicate_partial _percentage • sve_predicate_empty _percentage	• SVE_Effectiveness	• SVE
0x8075, SVE_PRED_EMPTY_SPEC	• sve_predicate_empty _percentage	• SVE_Effectiveness	• SVE
0x8076, SVE_PRED_FULL_SPEC	• sve_predicate_full _percentage	• SVE_Effectiveness	• SVE
0x8077, SVE_PRED_PARTIAL_SPEC	• sve_predicate_partial _percentage	• SVE_Effectiveness	• SVE
0x8079, SVE_PRED_NOT_FULL_SPEC	-	-	• SVE
0x8087, PRF_SPEC	-	-	• Spec_Operation
0x80BC, SVE_LDFF_SPEC	-	-	• SVE
0x80BD, SVE_LDFF_FAULT_SPEC	-	-	• SVE
0x80C0, FP_SCALE_OPS_SPEC	• sve_fp_ops_per_cycle • fp_ops_per_cycle	• FP_Arithmetic _Intensity	• FP_Operation
0x80C1, FP_FIXED_OPS_SPEC	• nonsve_fp_ops_per_cycle • fp_ops_per_cycle	• FP_Arithmetic _Intensity	• FP_Operation
0x80E3, ASE_SVE_INT8_SPEC	-	-	• SVE
0x80E7, ASE_SVE_INT16_SPEC	-	-	• SVE
0x80EB, ASE_SVE_INT32_SPEC	-	-	• SVE
0x80EF, ASE_SVE_INT64_SPEC	-	-	• SVE
0x810C, BR_INDNR_TAKEN_RETIRED	-	-	• Retired
0x8110, BR_IMMED_PRED_RETIRED	-	-	• Retired
0x8111, BR_IMMED_MIS_PRED _RETIRED	-	-	• Retired
0x8112, BR_IND_PRED_RETIRED	-	-	• Retired
0x8113, BR_IND_MIS_PRED _RETIRED	-	-	• Retired
0x8114, BR_RETURN_PRED_RETIRED	-	-	• Retired

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8115, BR_RETURN_MIS_PRED_RETIRED	-	-	• Retired
0x8116, BR_INDNR_PRED_RETIRED	-	-	• Retired
0x8117, BR_INDNR_MIS_PRED_RETIRED	-	-	• Retired
0x8118, BR_TAKEN_PRED_RETIRED	-	-	• Retired
0x8119, BR_TAKEN_MIS_PRED_RETIRED	-	-	• Retired
0x811A, BR_SKIP_PRED_RETIRED	-	-	• Retired
0x811B, BR_SKIP_MIS_PRED_RETIRED	-	-	• Retired
0x811C, BR_PRED_RETIRED	-	-	• Retired
0x811D, BR_IND_RETIRED	• branch_indirect_ratio	• Branch_Effectiveness	• Retired
0x811F, BRB_FILTRATE	-	-	• BRBE
0x8120, INST_FETCH_PERCYC	-	-	• Memory
0x8121, MEM_ACCESS_RD_PERCYC	-	-	• Memory
0x8124, INST_FETCH	-	-	• Memory
0x8128, DTLB_WALK_PERCYC	-	-	• TLB
0x8129, ITLB_WALK_PERCYC	-	-	• TLB
0x812A, SAMPLE_FEED_BR	-	-	• SPE
0x812B, SAMPLE_FEED_LD	-	-	• SPE
0x812C, SAMPLE_FEED_ST	-	-	• SPE
0x812D, SAMPLE_FEED_OP	-	-	• SPE
0x812E, SAMPLE_FEED_EVENT	-	-	• SPE
0x812F, SAMPLE_FEED_LAT	-	-	• SPE
0x8130, L1D_TLB_RW	-	-	• TLB
0x8131, L1I_TLB_RD	-	-	• TLB
0x8132, L1D_TLB_PRFM	-	-	• TLB
0x8133, L1I_TLB_PRFM	-	-	• TLB
0x8134, DTLB_HWUPD	-	-	• TLB
0x8135, ITLB_HWUPD	-	-	• TLB
0x8136, DTLB_STEP	-	-	• TLB
0x8137, ITLB_STEP	-	-	• TLB
0x8138, DTLB_WALK_LARGE	-	-	• TLB
0x8139, ITLB_WALK_LARGE	-	-	• TLB
0x813A, DTLB_WALK_SMALL	-	-	• TLB
0x813B, ITLB_WALK_SMALL	-	-	• TLB
0x813C, DTLB_WALK_RW	-	-	• TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x813D, ITLB_WALK_RD	-	-	• TLB
0x813E, DTLB_WALK_PRFM	-	-	• TLB
0x813F, ITLB_WALK_PRFM	-	-	• TLB
0x8140, L1D_CACHE_RW	-	-	• L1D_Cache
0x8141, L1I_CACHE_RD	-	-	• L1I_Cache
0x8142, L1D_CACHE_PRFM	-	-	• L1D_Cache
0x8143, L1I_CACHE_PRFM	-	-	• L1I_Cache
0x8144, L1D_CACHE_MISS	-	-	• L1D_Cache
0x8145, L1I_CACHE_HWPRF	-	-	• L1I_Cache
0x8146, L1D_CACHE_REFILL_PRFM	-	-	• L1D_Cache
0x8147, L1I_CACHE_REFILL_PRFM	-	-	• L1I_Cache
0x8148, L2D_CACHE_RW	-	-	• L2_Cache
0x814A, L2D_CACHE_PRFM	-	-	• L2_Cache
0x814C, L2D_CACHE_MISS	-	-	• L2_Cache
0x814E, L2D_CACHE_REFILL_PRFM	-	-	• L2_Cache
0x8154, L1D_CACHE_HWPRF	-	-	• L1D_Cache
0x8155, L2D_CACHE_HWPRF	-	-	• L2_Cache
0x8158, STALL_FRONTEND_MEMBOUND	• frontend_mem_bound • frontend_mem_cache_bound • frontend_mem_tlb_bound	• Topdown_Frontend	• Stall
0x8159, STALL_FRONTEND_L1I	• frontend_mem_cache_bound • frontend_cache_l1i_bound • frontend_cache_l2i_bound	• Topdown_Frontend	• Stall
0x815B, STALL_FRONTEND_MEM	• frontend_mem_cache_bound • frontend_cache_l1i_bound • frontend_cache_l2i_bound	• Topdown_Frontend	• Stall
0x815C, STALL_FRONTEND_TLB	• frontend_mem_tlb_bound	• Topdown_Frontend	• Stall
0x8160, STALL_FRONTEND_CPUBOUND	• frontend_core_bound • frontend_core_flush_bound • frontend_core_flow_bound	• Topdown_Frontend	• Stall
0x8161, STALL_FRONTEND_FLOW	• frontend_core_flow_bound	• Topdown_Frontend	• Stall
0x8162, STALL_FRONTEND_FLUSH	• frontend_bound • bad_speculation • frontend_core_flush_bound	• Topdown_Frontend • Topdown_L1	• Stall

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8164, STALL_BACKEND_MEMBOUND	• backend_mem_bound • backend_mem_cache_bound • backend_mem_tlb_bound • backend_mem_store_bound	• Topdown_Backend	• Stall
0x8165, STALL_BACKEND_L1D	• backend_mem_cache_bound • backend_cache_l1d_bound • backend_cache_l2d_bound	• Topdown_Backend	• Stall
0x8167, STALL_BACKEND_TLB	• backend_mem_tlb_bound	• Topdown_Backend	• Stall
0x8168, STALL_BACKEND_ST	• backend_mem_store_bound	• Topdown_Backend	• Stall
0x816A, STALL_BACKEND_CPUBOUND	• backend_core_bound • backend_core_rename_bound	• Topdown_Backend	• Stall
0x816B, STALL_BACKEND_BUSY	• backend_busy_bound	• Topdown_Backend	• Stall
0x816D, STALL_BACKEND_RENAME	• backend_core_rename_bound	• Topdown_Backend	• Stall
0x81BD, L2D_CACHE_REFILL_HWPRF	-	-	• L2_Cache
0x81C0, L1I_CACHE_HIT_RD	-	-	• L1I_Cache
0x81D0, L1I_CACHE_HIT_RD_FPRFM	-	-	• L1I_Cache
0x81E0, L1I_CACHE_HIT_RD_FHWPRF	-	-	• L1I_Cache
0x8200, L1I_CACHE_HIT	-	-	• L1I_Cache
0x8208, L1I_CACHE_HIT_PRFM	-	-	• L1I_Cache
0x8240, L1I_LFB_HIT_RD	-	-	• L1I_Cache
0x8250, L1I_LFB_HIT_RD_FPRFM	-	-	• L1I_Cache
0x8260, L1I_LFB_HIT_RD_FHWPRF	-	-	• L1I_Cache

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5. Metrics by metric group in Neoverse V3

Metrics are measured using different combinations of PMU events. They are organized into groups that can be analyzed together for a use case. To calculate the metrics, two or more PMU counters are programmed with the events listed for the metric. The counters are read at the same time to determine the metric value.

Summary:

- Total metrics: 67

Metrics for Neoverse V3 are grouped into the following metric groups:

- Topdown_L1, Topdown Level 1 (4 metrics)
- Topdown_Frontend, Topdown Frontend (8 metrics)
- Topdown_Backend, Topdown Backend (9 metrics)
- Cycle_Accounting, Cycle Accounting (2 metrics)
- General, General (1 metrics)
- MPKI, Misses Per Kilo Instructions (10 metrics)
- Miss_Ratio, Miss Ratio (10 metrics)
- SVE_Effectiveness, SVE Effectiveness (4 metrics)
- FP_Arithmetic_Intensity, Floating Point Arithmetic Intensity (3 metrics)
- FP_Precision_Mix, Floating Point Precision (3 metrics)
- Branch_Effectiveness, Branch Effectiveness (5 metrics)
- ITLB_Effectiveness, Instruction TLB Effectiveness (6 metrics)
- DTLB_Effectiveness, Data TLB Effectiveness (6 metrics)
- L1I_Cache_Effectiveness, L1 Instruction Cache Effectiveness (2 metrics)
- L1D_Cache_Effectiveness, L1 Data Cache Effectiveness (2 metrics)
- L2_Cache_Effectiveness, L2 Unified Cache Effectiveness (2 metrics)
- LL_Cache_Effectiveness, Last Level Cache Effectiveness (3 metrics)
- Operation_Mix, Speculative Operation Mix (9 metrics)

5.1 Topdown_L1 metrics for Neoverse V3

Topdown Level 1. This metric group contains the first set of metrics to begin topdown analysis of application performance, which provide the percentage distribution of processor pipeline utilization.

Summary of metrics in Topdown_L1:

- Total metrics: 4

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 5-1: Topdown_L1 metrics summary

Metric	Name	Description
backend_bound	Backend Bound	This metric is the percentage of total slots that were stalled due to resource constraints in the...
bad_speculation	Bad Speculation	This metric is the percentage of total slots that executed operations and didn't retire due to a...
frontend_bound	Frontend Bound	This metric is the percentage of total slots that were stalled due to resource constraints in the...
retiring	Retiring	This metric is the percentage of total slots that retired operations, which indicates cycles that...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

backend_bound, Backend Bound, metric

This metric is the percentage of total slots that were stalled due to resource constraints in the backend of the processor.

Units

This unit is expressed as percent of slots.

Formula

STALL_SLOT_BACKEND / (10 * CPU_CYCLES) * 100

Related telemetry artifacts

Events

CPU_CYCLES
STALL_SLOT_BACKEND

Metric group

Topdown_L1

Methodology

Stage 1

bad_speculation, Bad Speculation, metric

This metric is the percentage of total slots that executed operations and didn't retire due to a pipeline flush. This indicates cycles that were utilized but inefficiently.

Units

This unit is expressed as percent of slots.

Formula

(1 - STALL_SLOT / (10 * CPU_CYCLES)) * (1 - OP_RETIRED / OP_SPEC) * 100 + STALL_FRONTEND_FLUSH / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
OP_RETIRED
OP_SPEC
STALL_FRONTEND_FLUSH

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

STALL_SLOT

Metric group

Topdown_L1

Methodology

Stage 1

frontend_bound, Frontend Bound, metric

This metric is the percentage of total slots that were stalled due to resource constraints in the frontend of the processor.

Units

This unit is expressed as percent of slots.

Formula

(STALL_SLOT_FRONTEND / (10 * CPU_CYCLES) - STALL_FRONTEND_FLUSH / CPU_CYCLES) * 100

Related telemetry artifacts

Events

CPU_CYCLES
STALL_FRONTEND_FLUSH
STALL_SLOT_FRONTEND

Metric group

Topdown_L1

Methodology

Stage 1

retiring, Retiring, metric

This metric is the percentage of total slots that retired operations, which indicates cycles that were utilized efficiently.

Units

This unit is expressed as percent of slots.

Formula

(1 - STALL_SLOT / (CPU_CYCLES * 10)) * (OP_RETIRED / OP_SPEC) * 100

Related telemetry artifacts

Events

CPU_CYCLES
OP_RETIRED
OP_SPEC
STALL_SLOT

Metric group

Topdown_L1

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Methodology

Stage 1

5.2 Topdown_Frontend metrics for Neoverse V3

Topdown Frontend. This metric group contains a set of metrics to analyse a frontend bound workload.

Summary of metrics in Topdown_Frontend:

- Total metrics: 8

Table 5-2: Topdown_Frontend metrics summary

Metric	Name	Description
frontend_cache_l1i_bound	Frontend Cache L1I Bound	This metric is the percentage of total cycles stalled in the frontend due to memory access...
frontend_cache_l2i_bound	Frontend Cache L2I Bound	This metric is the percentage of total cycles stalled in the frontend due to memory access...
frontend_core_bound	Frontend Core Bound	This metric is the percentage of total cycles stalled in the frontend due to frontend core...
frontend_core_flow_bound	Frontend Core Flow Bound	This metric is the percentage of total cycles stalled in the frontend as the decode unit is...
frontend_core_flush_bound	Frontend Core Flush Bound	This metric is the percentage of total cycles stalled in the frontend as the processor is...
frontend_mem_bound	Frontend Memory Bound	This metric is the percentage of total cycles stalled in the frontend due to frontend core...
frontend_mem_cache_bound	Frontend Mem Cache Bound	This metric is the percentage of total cycles stalled in the frontend due to instruction fetch...
frontend_mem_tlb_bound	Frontend Mem TLB Bound	This metric is the percentage of total cycles stalled in the frontend due to instruction fetch...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

frontend_cache_l1i_bound, Frontend Cache L1I Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to memory access latency issues caused by level 1 instruction cache misses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_L1I / (STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) * 100

Related telemetry artifacts

Events

STALL_FRONTEND_L1I

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

STALL_FRONTEND_MEM

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_cache_l2i_bound, Frontend Cache L2I Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to memory access latency issues caused by level 2 instruction cache misses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_MEM / (STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) * 100

Related telemetry artifacts

Events

STALL_FRONTEND_L1I
STALL_FRONTEND_MEM

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_core_bound, Frontend Core Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to frontend core resource constraints not related to instruction fetch latency issues caused by memory access components.

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_CPUBOUND / STALL_FRONTEND * 100

Related telemetry artifacts

Events

STALL_FRONTEND
STALL_FRONTEND_CPUBOUND

Metric group

Topdown_Frontend

Methodology

Stage 1

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

frontend_core_flow_bound, Frontend Core Flow Bound, metric

This metric is the percentage of total cycles stalled in the frontend as the decode unit is awaiting input from the branch prediction unit.

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_FLOW / STALL_FRONTEND_CPUBOUND * 100

Related telemetry artifacts

Events

STALL_FRONTEND_CPUBOUND
STALL_FRONTEND_FLOW

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_core_flush_bound, Frontend Core Flush Bound, metric

This metric is the percentage of total cycles stalled in the frontend as the processor is recovering from a pipeline flush caused by bad speculation or other machine resteers.

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_FLUSH / STALL_FRONTEND_CPUBOUND * 100

Related telemetry artifacts

Events

STALL_FRONTEND_CPUBOUND
STALL_FRONTEND_FLUSH

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_mem_bound, Frontend Memory Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to frontend core resource constraints related to the instruction fetch latency issues caused by memory access components.

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_MEMBOUND / STALL_FRONTEND * 100

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Events

STALL_FRONTEND
STALL_FRONTEND_MEMBOUND

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_mem_cache_bound, Frontend Mem Cache Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to instruction fetch latency issues caused by instruction cache misses.

Units

This unit is expressed as percent of cycles.

Formula

(STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) / STALL_FRONTEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_FRONTEND_L1I
STALL_FRONTEND_MEM
STALL_FRONTEND_MEMBOUND

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_mem_tlb_bound, Frontend Mem TLB Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to instruction fetch latency issues caused by instruction TLB misses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_TLB / STALL_FRONTEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_FRONTEND_MEMBOUND
STALL_FRONTEND_TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric group

Topdown_Frontend

Methodology

Stage 1

5.3 Topdown_Backend metrics for Neoverse V3

Topdown Backend. This metric group contains a set of metrics to analyze a backend bound workload.

Summary of metrics in Topdown_Backend:

- Total metrics: 9

Table 5-3: Topdown_Backend metrics summary

Metric	Name	Description
backend_busy_bound	Backend Busy Bound	This metric is the percentage of total cycles stalled in the backend due to issue queues being...
backend_cache_l1d_bound	Backend Cache L1D Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...
backend_cache_l2d_bound	Backend Cache L2D Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...
backend_core_bound	Backend Core Bound	This metric is the percentage of total cycles stalled in the backend due to backend core resource...
backend_core_rename_bound	Backend Core Rename Bound	This metric is the percentage of total cycles stalled in the backend as the rename unit registers...
backend_mem_bound	Backend Memory Bound	This metric is the percentage of total cycles stalled in the backend due to backend core resource...
backend_mem_cache_bound	Backend Memory Cache Bound	This metric is the percentage of total cycles stalled in the backend due to memory latency issues...
backend_mem_store_bound	Backend Memory Store Bound	This metric is the percentage of total cycles stalled in the backend due to memory write pending...
backend_mem_tlb_bound	Backend Memory TLB Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

backend_busy_bound, Backend Busy Bound, metric

This metric is the percentage of total cycles stalled in the backend due to issue queues being full to accept operations for execution.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_BUSY / STALL_BACKEND * 100

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Events

STALL_BACKEND
STALL_BACKEND_BUSY

Metric group

Topdown_Backend

Methodology

Stage 1

backend_cache_l1d_bound, Backend Cache L1D Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory access latency issues caused by level 1 data cache misses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_L1D / (STALL_BACKEND_L1D + STALL_BACKEND_MEM) * 100

Related telemetry artifacts

Events

STALL_BACKEND_L1D
STALL_BACKEND_MEM

Metric group

Topdown_Backend

Methodology

Stage 1

backend_cache_l2d_bound, Backend Cache L2D Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory access latency issues caused by level 2 data cache misses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_MEM / (STALL_BACKEND_L1D + STALL_BACKEND_MEM) * 100

Related telemetry artifacts

Events

STALL_BACKEND_L1D
STALL_BACKEND_MEM

Metric group

Topdown_Backend

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Methodology

Stage 1

backend_core_bound, Backend Core Bound, metric

This metric is the percentage of total cycles stalled in the backend due to backend core resource constraints not related to instruction fetch latency issues caused by memory access components.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_CPUBOUND / STALL_BACKEND * 100

Related telemetry artifacts

Events

STALL_BACKEND
STALL_BACKEND_CPUBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_core_rename_bound, Backend Core Rename Bound, metric

This metric is the percentage of total cycles stalled in the backend as the rename unit registers are unavailable.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_RENAME / STALL_BACKEND_CPUBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_CPUBOUND
STALL_BACKEND_RENAME

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_bound, Backend Memory Bound, metric

This metric is the percentage of total cycles stalled in the backend due to backend core resource constraints related to memory access latency issues caused by memory access components.

Units

This unit is expressed as percent of cycles.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Formula

STALL_BACKEND_MEMBOUND / STALL_BACKEND * 100

Related telemetry artifacts

Events

STALL_BACKEND
STALL_BACKEND_MEMBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_cache_bound, Backend Memory Cache Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory latency issues caused by data cache misses.

Units

This unit is expressed as percent of cycles.

Formula

(STALL_BACKEND_L1D + STALL_BACKEND_MEM) / STALL_BACKEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_L1D
STALL_BACKEND_MEM
STALL_BACKEND_MEMBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_store_bound, Backend Memory Store Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory write pending caused by stores stalled in the pre-commit stage.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_ST / STALL_BACKEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEMBOUND

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

STALL_BACKEND_ST

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_tlb_bound, Backend Memory TLB Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory access latency issues caused by data TLB misses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_TLB / STALL_BACKEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEMBOUND
STALL_BACKEND_TLB

Metric group

Topdown_Backend

Methodology

Stage 1

5.4 Cycle_Accounting metrics for Neoverse V3

Cycle Accounting. This metric group contains a set of metrics that measure the percentage of processor cycles stalled in either frontend or backend of the processor.

Summary of metrics in Cycle_Accounting:

- Total metrics: 2

Table 5-4: Cycle_Accounting metrics summary

Metric	Name	Description
backend_stalled_cycles	Backend Stalled Cycles	This metric is the percentage of cycles that were stalled due to resource constraints in the...
frontend_stalled_cycles	Frontend Stalled Cycles	This metric is the percentage of cycles that were stalled due to resource constraints in the...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

backend_stalled_cycles, Backend Stalled Cycles, metric

This metric is the percentage of cycles that were stalled due to resource constraints in the backend unit of the processor.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
STALL_BACKEND

Metric group

Cycle_Accounting

Methodology

Stage 2

frontend_stalled_cycles, Frontend Stalled Cycles, metric

This metric is the percentage of cycles that were stalled due to resource constraints in the frontend unit of the processor.

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
STALL_FRONTEND

Metric group

Cycle_Accounting

Methodology

Stage 2

5.5 General metrics for Neoverse V3

General. This metric group contains general CPU metrics for performance analysis.

Summary of metrics in General:

- Total metrics: 1

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 5-5: General metrics summary

Metric	Name	Description
ipc	Instructions Per Cycle	This metric measures the number of instructions retired per cycle.

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

ipc, Instructions Per Cycle, metric

This metric measures the number of instructions retired per cycle.

Units

This unit is expressed as per cycle.

Formula

INST_RETIRED / CPU_CYCLES

Related telemetry artifacts

Events

CPU_CYCLES
INST_RETIRED

Metric group

General

Methodology

Stage 2

5.6 MPKI metrics for Neoverse V3

Misses Per Kilo Instructions. This metric group contains metrics for different CPU resources that can be measured as misses per kilo instructions.

Summary of metrics in MPKI:

- Total metrics: 10

Table 5-6: MPKI metrics summary

Metric	Name	Description
branch_mpki	Branch MPKI	This metric measures the number of branch mispredictions per thousand instructions executed.
dtlb_mpki	DTLB MPKI	This metric measures the number of data TLB Walks per thousand instructions executed.
itlb_mpki	ITLB MPKI	This metric measures the number of instruction TLB Walks per thousand instructions executed.
l1d_cache_mpki	L1D Cache MPKI	This metric measures the number of level 1 data cache accesses missed per thousand instructions...
l1d_tlb_mpki	L1 Data TLB MPKI	This metric measures the number of level 1 data TLB accesses missed per thousand instructions...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric	Name	Description
l1i_cache_mpki	L1I Cache MPKI	This metric measures the number of level 1 instruction cache accesses missed per thousand...
l1i_tlb_mpki	L1 Instruction TLB MPKI	This metric measures the number of level 1 instruction TLB accesses missed per thousand...
l2_cache_mpki	L2 Cache MPKI	This metric measures the number of level 2 unified cache accesses missed per thousand...
l2_tlb_mpki	L2 Unified TLB MPKI	This metric measures the number of level 2 unified TLB accesses missed per thousand instructions...
ll_cache_read_mpki	LL Cache Read MPKI	This metric measures the number of last level cache read accesses missed per thousand...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

branch_mpki, Branch MPKI, metric

This metric measures the number of branch mispredictions per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

BR_MIS_PRED_RETIRED / INST_RETIRED * 1000

Related telemetry artifacts

Events

BR_MIS_PRED_RETIRED
INST_RETIRED

Metric group

MPKI

Other metric group: Branch_Effectiveness

Methodology

Stage 2

dtlb_mpki, DTLB MPKI, metric

This metric measures the number of data TLB Walks per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

DTLB_WALK / INST_RETIRED * 1000

Related telemetry artifacts

Events

DTLB_WALK
INST_RETIRED

Metric group

MPKI

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Other metric group: DTLB_Effectiveness

Methodology

Stage 2

itlb_mpki, ITLB MPKI, metric

This metric measures the number of instruction TLB Walks per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

ITLB_WALK / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

ITLB_WALK

Metric group

MPKI

Other metric group: ITLB_Effectiveness

Methodology

Stage 2

l1d_cache_mpki, L1D Cache MPKI, metric

This metric measures the number of level 1 data cache accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L1D_CACHE_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

L1D_CACHE_REFILL

Metric group

MPKI

Other metric group: L1D_Cache_Effectiveness

Methodology

Stage 2

l1d_tlb_mpki, L1 Data TLB MPKI, metric

This metric measures the number of level 1 data TLB accesses missed per thousand instructions executed.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Units

This unit is expressed as mpki.

Formula

L1D_TLB_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

L1D_TLB_REFILL

Metric group

MPKI

Other metric group: DTLB_Effectiveness

Methodology

Stage 2

l1i_cache_mpki, L1I Cache MPKI, metric

This metric measures the number of level 1 instruction cache accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L1I_CACHE_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

L1I_CACHE_REFILL

Metric group

MPKI

Other metric group: L1I_Cache_Effectiveness

Methodology

Stage 2

l1i_tlb_mpki, L1 Instruction TLB MPKI, metric

This metric measures the number of level 1 instruction TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L1I_TLB_REFILL / INST_RETIRED * 1000

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Events

INST_RETIRED

L1I_TLB_REFILL

Metric group

MPKI

Other metric group: ITLB_Effectiveness

Methodology

Stage 2

l2_cache_mpki, L2 Cache MPKI, metric

This metric measures the number of level 2 unified cache accesses missed per thousand instructions executed. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a unified cache.

Units

This unit is expressed as mpki.

Formula

L2D_CACHE_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

L2D_CACHE_REFILL

Metric group

MPKI

Other metric group: L2_Cache_Effectiveness

Methodology

Stage 2

l2_tlb_mpki, L2 Unified TLB MPKI, metric

This metric measures the number of level 2 unified TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L2D_TLB_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

L2D_TLB_REFILL

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric group

MPKI

Other metric group: DTLB_Effectiveness

Other metric group: ITLB_Effectiveness

Methodology

Stage 2

ll_cache_read_mpki, LL Cache Read MPKI, metric

This metric measures the number of last level cache read accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

LL_CACHE_MISS_RD / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

LL_CACHE_MISS_RD

Metric group

MPKI

Other metric group: LL_Cache_Effectiveness

Methodology

Stage 2

5.7 Miss_Ratio metrics for Neoverse V3

Miss Ratio. This metric group contains metrics to measure miss ratios of different processor resources.

Summary of metrics in Miss_Ratio:

- Total metrics: 10

Table 5-7: Miss_Ratio metrics summary

Metric	Name	Description
branch_misprediction_ratio	Branch Misprediction Ratio	This metric measures the ratio of branches mispredicted to the total number of branches...
dtlb_walk_ratio	DTLB Walk Ratio	This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This...
itlb_walk_ratio	ITLB Walk Ratio	This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric	Name	Description
l1d_cache_miss_ratio	L1D Cache Miss Ratio	This metric measures the ratio of level 1 data cache accesses missed to the total number of level...
l1d_tlb_miss_ratio	L1 Data TLB Miss Ratio	This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1...
l1i_cache_miss_ratio	L1I Cache Miss Ratio	This metric measures the ratio of level 1 instruction cache accesses missed to the total number...
l1i_tlb_miss_ratio	L1 Instruction TLB Miss Ratio	This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of...
l2_cache_miss_ratio	L2 Cache Miss Ratio	This metric measures the ratio of level 2 cache accesses missed to the total number of level 2...
l2_tlb_miss_ratio	L2 Unified TLB Miss Ratio	This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...
ll_cache_read_miss_ratio	LL Cache Read Miss Ratio	This metric measures the ratio of last level cache read accesses missed to the total number of...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

branch_misprediction_ratio, Branch Misprediction Ratio, metric

This metric measures the ratio of branches mispredicted to the total number of branches architecturally executed. This gives an indication of the effectiveness of the branch prediction unit.

Units

This unit is expressed as per branch.

Formula

BR_MIS_PRED_RETIRED / BR_RETIRED

Related telemetry artifacts

Events

BR_MIS_PRED_RETIRED
BR_RETIRED

Metric group

Miss_Ratio
Other metric group: Branch_Effectiveness

Methodology

Stage 2

dtlb_walk_ratio, DTLB Walk Ratio, metric

This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This gives an indication of the effectiveness of the data TLB accesses.

Units

This unit is expressed as per tlb access.

Formula

DTLB_WALK / L1D_TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Events

DTLB_WALK
L1D_TLB

Metric group

Miss_Ratio
Other metric group: DTLB_Effectiveness

Methodology

Stage 2

itlb_walk_ratio, ITLB Walk Ratio, metric

This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB accesses. This gives an indication of the effectiveness of the instruction TLB accesses.

Units

This unit is expressed as per tlb access.

Formula

ITLB_WALK / L1I_TLB

Related telemetry artifacts

Events

ITLB_WALK
L1I_TLB

Metric group

Miss_Ratio
Other metric group: ITLB_Effectiveness

Methodology

Stage 2

l1d_cache_miss_ratio, L1D Cache Miss Ratio, metric

This metric measures the ratio of level 1 data cache accesses missed to the total number of level 1 data cache accesses. This gives an indication of the effectiveness of the level 1 data cache.

Units

This unit is expressed as per cache access.

Formula

L1D_CACHE_REFILL / L1D_CACHE

Related telemetry artifacts

Events

L1D_CACHE
L1D_CACHE_REFILL

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric group

Miss_Ratio

Other metric group: L1D_Cache_Effectiveness

Methodology

Stage 2

l1d_tlb_miss_ratio, L1 Data TLB Miss Ratio, metric

This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1 data TLB accesses. This gives an indication of the effectiveness of the level 1 data TLB.

Units

This unit is expressed as per tlb access.

Formula

L1D_TLB_REFILL / L1D_TLB

Related telemetry artifacts

Events

L1D_TLB

L1D_TLB_REFILL

Metric group

Miss_Ratio

Other metric group: DTLB_Effectiveness

Methodology

Stage 2

l1i_cache_miss_ratio, L1I Cache Miss Ratio, metric

This metric measures the ratio of level 1 instruction cache accesses missed to the total number of level 1 instruction cache accesses. This gives an indication of the effectiveness of the level 1 instruction cache.

Units

This unit is expressed as per cache access.

Formula

L1I_CACHE_REFILL / L1I_CACHE

Related telemetry artifacts

Events

L1I_CACHE

L1I_CACHE_REFILL

Metric group

Miss_Ratio

Other metric group: L1I_Cache_Effectiveness

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Methodology

Stage 2

l1i_tlb_miss_ratio, L1 Instruction TLB Miss Ratio, metric

This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of level 1 instruction TLB accesses. This gives an indication of the effectiveness of the level 1 instruction TLB.

Units

This unit is expressed as per tlb access.

Formula

L1I_TLB_REFILL / L1I_TLB

Related telemetry artifacts

Events

L1I_TLB

L1I_TLB_REFILL

Metric group

Miss_Ratio

Other metric group: ITLB_Effectiveness

Methodology

Stage 2

l2_cache_miss_ratio, L2 Cache Miss Ratio, metric

This metric measures the ratio of level 2 cache accesses missed to the total number of level 2 cache accesses. This gives an indication of the effectiveness of the level 2 cache, which is a unified cache that stores both data and instruction. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a unified cache.

Units

This unit is expressed as per cache access.

Formula

L2D_CACHE_REFILL / L2D_CACHE

Related telemetry artifacts

Events

L2D_CACHE

L2D_CACHE_REFILL

Metric group

Miss_Ratio

Other metric group: L2_Cache_Effectiveness

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

l2_tlb_miss_ratio, L2 Unified TLB Miss Ratio, metric

This metric measures the ratio of level 2 unified TLB accesses missed to the total number of level 2 unified TLB accesses. This gives an indication of the effectiveness of the level 2 TLB.

Units

This unit is expressed as per tlb access.

Formula

L2D_TLB_REFILL / L2D_TLB

Related telemetry artifacts

Events

L2D_TLB
L2D_TLB_REFILL

Metric group

Miss_Ratio
Other metric group: DTLB_Effectiveness
Other metric group: ITLB_Effectiveness

Methodology

Stage 2

ll_cache_read_miss_ratio, LL Cache Read Miss Ratio, metric

This metric measures the ratio of last level cache read accesses missed to the total number of last level cache accesses. This gives an indication of the effectiveness of the last level cache for read traffic. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a system level cache.

Units

This unit is expressed as per cache access.

Formula

LL_CACHE_MISS_RD / LL_CACHE_RD

Related telemetry artifacts

Events

LL_CACHE_MISS_RD
LL_CACHE_RD

Metric group

Miss_Ratio
Other metric group: LL_Cache_Effectiveness

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.8 SVE_Effectiveness metrics for Neoverse V3

SVE Effectiveness. This metric group contains metrics to evaluate the effectiveness of predicated SVE instruction execution on this processor.

Summary of metrics in SVE_Effectiveness:

- Total metrics: 4

Table 5-8: SVE_Effectiveness metrics summary

Metric	Name	Description
sve_predicate_empty_percentage	SVE Empty Predicate Percentage	This metric measures scalable vector operations with no active predicates as a percentage of sve...
sve_predicate_full_percentage	SVE Full Predicate Percentage	This metric measures scalable vector operations with all active predicates as a percentage of sve...
sve_predicate_partial_percentage	SVE Partial Predicate Percentage	This metric measures scalable vector operations with at least one active predicates as a...
sve_predicate_percentage	SVE Predicate Percentage	This metric measures scalable vector operations with predicates as a percentage of operations...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

sve_predicate_empty_percentage, SVE Empty Predicate Percentage, metric

This metric measures scalable vector operations with no active predicates as a percentage of sve predicated operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

SVE_PRED_EMPTY_SPEC / SVE_PRED_SPEC * 100

Related telemetry artifacts

Events

SVE_PRED_EMPTY_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

sve_predicate_full_percentage, SVE Full Predicate Percentage, metric

This metric measures scalable vector operations with all active predicates as a percentage of sve predicated operations speculatively executed.

Units

This unit is expressed as percent of operations.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Formula

SVE_PRED_FULL_SPEC / SVE_PRED_SPEC * 100

Related telemetry artifacts

Events

SVE_PRED_FULL_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

sve_predicate_partial_percentage, SVE Partial Predicate Percentage, metric

This metric measures scalable vector operations with at least one active predicates as a percentage of sve predicated operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

SVE_PRED_PARTIAL_SPEC / SVE_PRED_SPEC * 100

Related telemetry artifacts

Events

SVE_PRED_PARTIAL_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

sve_predicate_percentage, SVE Predicate Percentage, metric

This metric measures scalable vector operations with predicates as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

SVE_PRED_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC
SVE_PRED_SPEC

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric group

SVE_Effectiveness

Methodology

Stage 2

5.9 FP_Arithmetic_Intensity metrics for Neoverse V3

Floating Point Arithmetic Intensity. This metric group contains metrics to evaluate the effectiveness of floating point instruction execution on this processor.

Summary of metrics in FP_Arithmetic_Intensity:

- Total metrics: 3

Table 5-9: FP_Arithmetic_Intensity metrics summary

Metric	Name	Description
fp_ops_per_cycle	Floating Point Operations per Cycle	This metric measures floating point operations per cycle in any precision performed by any...
nonsve_fp_ops_per_cycle	Non-SVE Floating Point Operations per Cycle	This metric measures floating point operations per cycle in any precision performed by an...
sve_fp_ops_per_cycle	SVE Floating Point Operations per Cycle	This metric measures floating point operations per cycle in any precision performed by SVE...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

fp_ops_per_cycle, Floating Point Operations per Cycle, metric

This metric measures floating point operations per cycle in any precision performed by any instruction. Operations are counted by computation and by vector lanes, fused computations such as multiply-add count as twice per vector lane for example.

Units

This unit is expressed as operations per cycle.

Formula

(FP_SCALE_OPS_SPEC + FP_FIXED_OPS_SPEC) / CPU_CYCLES

Related telemetry artifacts

Events

CPU_CYCLES
FP_FIXED_OPS_SPEC
FP_SCALE_OPS_SPEC

Metric group

FP_Arithmetic_Intensity

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Methodology

Stage 2

nonsve_fp_ops_per_cycle, Non-SVE Floating Point Operations per Cycle, metric

This metric measures floating point operations per cycle in any precision performed by an instruction that is not an SVE instruction. Operations are counted by computation and by vector lanes, fused computations such as multiply-add count as twice per vector lane for example.

Units

This unit is expressed as operations per cycle.

Formula

FP_FIXED_OPS_SPEC / CPU_CYCLES

Related telemetry artifacts

Events

CPU_CYCLES
FP_FIXED_OPS_SPEC

Metric group

FP_Arithmetic_Intensity

Methodology

Stage 2

sve_fp_ops_per_cycle, SVE Floating Point Operations per Cycle, metric

This metric measures floating point operations per cycle in any precision performed by SVE instructions. Operations are counted by computation and by vector lanes, fused computations such as multiply-add count as twice per vector lane for example.

Units

This unit is expressed as operations per cycle.

Formula

FP_SCALE_OPS_SPEC / CPU_CYCLES

Related telemetry artifacts

Events

CPU_CYCLES
FP_SCALE_OPS_SPEC

Metric group

FP_Arithmetic_Intensity

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.10 FP_Precision_Mix metrics for Neoverse V3

Floating Point Precision. This metric group contains metrics to evaluate the precision of floating point instruction execution on this processor.

Summary of metrics in FP_Precision_Mix:

- Total metrics: 3

Table 5-10: FP_Precision_Mix metrics summary

Metric	Name	Description
fp16_percentage	Half Precision Floating Point Percentage	This metric measures half-precision floating point operations as a percentage of operations...
fp32_percentage	Single Precision Floating Point Percentage	This metric measures single-precision floating point operations as a percentage of operations...
fp64_percentage	Double Precision Floating Point Percentage	This metric measures double-precision floating point operations as a percentage of operations...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

fp16_percentage, Half Precision Floating Point Percentage, metric

This metric measures half-precision floating point operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

FP_HP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

FP_HP_SPEC
INST_SPEC

Metric group

FP_Precision_Mix

Methodology

Stage 2

fp32_percentage, Single Precision Floating Point Percentage, metric

This metric measures single-precision floating point operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Formula

FP_SP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

FP_SP_SPEC

INST_SPEC

Metric group

FP_Precision_Mix

Methodology

Stage 2

fp64_percentage, Double Precision Floating Point Percentage, metric

This metric measures double-precision floating point operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

FP_DP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

FP_DP_SPEC

INST_SPEC

Metric group

FP_Precision_Mix

Methodology

Stage 2

5.11 Branch_Effectiveness metrics for Neoverse V3

Branch Effectiveness. This metric group contains metrics to evaluate the effectiveness of branch instruction execution on this processor.

Summary of metrics in Branch_Effectiveness:

- Total metrics: 5

Table 5-11: Branch_Effectiveness metrics summary

Metric	Name	Description
branch_direct_ratio	Branch Direct Ratio	This metric measures the ratio of direct branches retired to the total number of branches...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric	Name	Description
branch_indirect_ratio	Branch Indirect Ratio	This metric measures the ratio of indirect branches retired, including function returns, to the...
branch_misprediction_ratio	Branch Misprediction Ratio	This metric measures the ratio of branches mispredicted to the total number of branches...
branch_mpki	Branch MPKI	This metric measures the number of branch mispredictions per thousand instructions executed.
branch_return_ratio	Branch Return Ratio	This metric measures the ratio of branches retired that are function returns to the total number...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

branch_direct_ratio, Branch Direct Ratio, metric

This metric measures the ratio of direct branches retired to the total number of branches architecturally executed.

Units

This unit is expressed as per branch.

Formula

BR_IMMED_RETIRED / BR_RETIRED

Related telemetry artifacts

Events

BR_IMMED_RETIRED
BR_RETIRED

Metric group

Branch_Effectiveness

Methodology

Stage 2

branch_indirect_ratio, Branch Indirect Ratio, metric

This metric measures the ratio of indirect branches retired, including function returns, to the total number of branches architecturally executed.

Units

This unit is expressed as per branch.

Formula

BR_IND_RETIRED / BR_RETIRED

Related telemetry artifacts

Events

BR_IND_RETIRED
BR_RETIRED

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric group

Branch_Effectiveness

Methodology

Stage 2

branch_misprediction_ratio**, Branch Misprediction Ratio, metric

This metric measures the ratio of branches mispredicted to the total number of branches architecturally executed. This gives an indication of the effectiveness of the branch prediction unit.

Units

This unit is expressed as per branch.

Formula

BR_MIS_PRED_RETIRED / BR_RETIRED

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

BR_MIS_PRED_RETIRED
BR_RETIRED

Metric group

Branch_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

branch_mpki**, Branch MPKI, metric

This metric measures the number of branch mispredictions per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

BR_MIS_PRED_RETIRED / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

BR_MIS_PRED_RETIRED
INST_RETIRED

Metric group

Branch_Effectiveness
Other metric group: MPKI

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Methodology

Stage 2

branch_return_ratio, Branch Return Ratio, metric

This metric measures the ratio of branches retired that are function returns to the total number of branches architecturally executed.

Units

This unit is expressed as per branch.

Formula

BR_RETURN_RETIRED / BR_RETIRED

Related telemetry artifacts

Events

BR_RETIRED
BR_RETURN_RETIRED

Metric group

Branch_Effectiveness

Methodology

Stage 2

5.12 ITLB_Effectiveness metrics for Neoverse V3

Instruction TLB Effectiveness. This metric group contains metrics to evaluate the effectiveness of instruction TLB on this processor.

Summary of metrics in ITLB_Effectiveness:

- Total metrics: 6

Table 5-12: ITLB_Effectiveness metrics summary

Metric	Name	Description
itlb_mpki	ITLB MPKI	This metric measures the number of instruction TLB Walks per thousand instructions executed.
itlb_walk_ratio	ITLB Walk Ratio	This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB...
l1i_tlb_miss_ratio	L1 Instruction TLB Miss Ratio	This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of...
l1i_tlb_mpki	L1 Instruction TLB MPKI	This metric measures the number of level 1 instruction TLB accesses missed per thousand...
l2_tlb_miss_ratio	L2 Unified TLB Miss Ratio	This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...
l2_tlb_mpki	L2 Unified TLB MPKI	This metric measures the number of level 2 unified TLB accesses missed per thousand instructions...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

itlb_mpki**, ITLB MPKI, metric

This metric measures the number of instruction TLB Walks per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

ITLB_WALK / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED
ITLB_WALK

Metric group

ITLB_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

itlb_walk_ratio**, ITLB Walk Ratio, metric

This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB accesses. This gives an indication of the effectiveness of the instruction TLB accesses.

Units

This unit is expressed as per tlb access.

Formula

ITLB_WALK / L1I_TLB

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

ITLB_WALK
L1I_TLB

Metric group

ITLB_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

l1i_tlb_miss_ratio**, L1 Instruction TLB Miss Ratio, metric

This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of level 1 instruction TLB accesses. This gives an indication of the effectiveness of the level 1 instruction TLB.

Units

This unit is expressed as per tlb access.

Formula

L1I_TLB_REFILL / L1I_TLB

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

L1I_TLB
L1I_TLB_REFILL

Metric group

ITLB_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l1i_tlb_mpki**, L1 Instruction TLB MPKI, metric

This metric measures the number of level 1 instruction TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L1I_TLB_REFILL / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED
L1I_TLB_REFILL

Metric group

ITLB_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

l2_tlb_miss_ratio**, L2 Unified TLB Miss Ratio, metric

This metric measures the ratio of level 2 unified TLB accesses missed to the total number of level 2 unified TLB accesses. This gives an indication of the effectiveness of the level 2 TLB.

Units

This unit is expressed as per tlb access.

Formula

L2D_TLB_REFILL / L2D_TLB

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

L2D_TLB
L2D_TLB_REFILL

Metric group

ITLB_Effectiveness
Other metric group: DTLB_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l2_tlb_mpki**, L2 Unified TLB MPKI, metric

This metric measures the number of level 2 unified TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L2D_TLB_REFILL / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED
L2D_TLB_REFILL

Metric group

ITLB_Effectiveness
Other metric group: DTLB_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.13 DTLB_Effectiveness metrics for Neoverse V3

Data TLB Effectiveness. This metric group contains metrics to evaluate the effectiveness of data TLB on this processor.

Summary of metrics in DTLB_Effectiveness:

- Total metrics: 6

Table 5-13: DTLB_Effectiveness metrics summary

Metric	Name	Description
dtlb_mpki	DTLB MPKI	This metric measures the number of data TLB Walks per thousand instructions executed.
dtlb_walk_ratio	DTLB Walk Ratio	This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This...
l1d_tlb_miss_ratio	L1 Data TLB Miss Ratio	This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1...
l1d_tlb_mpki	L1 Data TLB MPKI	This metric measures the number of level 1 data TLB accesses missed per thousand instructions...
l2_tlb_miss_ratio	L2 Unified TLB Miss Ratio	This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...
l2_tlb_mpki	L2 Unified TLB MPKI	This metric measures the number of level 2 unified TLB accesses missed per thousand instructions...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

dtlb_mpki**, DTLB MPKI, metric

This metric measures the number of data TLB Walks per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

DTLB_WALK / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

DTLB_WALK
INST_RETIRED

Metric group

DTLB_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

dtlb_walk_ratio**, DTLB Walk Ratio, metric

This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This gives an indication of the effectiveness of the data TLB accesses.

Units

This unit is expressed as per tlb access.

Formula

DTLB_WALK / L1D_TLB

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

DTLB_WALK
L1D_TLB

Metric group

DTLB_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l1d_tlb_miss_ratio**, L1 Data TLB Miss Ratio, metric

This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1 data TLB accesses. This gives an indication of the effectiveness of the level 1 data TLB.

Units

This unit is expressed as per tlb access.

Formula

L1D_TLB_REFILL / L1D_TLB

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

L1D_TLB
L1D_TLB_REFILL

Metric group

DTLB_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

l1d_tlb_mpki**, L1 Data TLB MPKI, metric

This metric measures the number of level 1 data TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L1D_TLB_REFILL / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED
L1D_TLB_REFILL

Metric group

DTLB_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

l2_tlb_miss_ratio***, L2 Unified TLB Miss Ratio, metric

This metric measures the ratio of level 2 unified TLB accesses missed to the total number of level 2 unified TLB accesses. This gives an indication of the effectiveness of the level 2 TLB.

Units

This unit is expressed as per tlb access.

Formula

L2D_TLB_REFILL / L2D_TLB

*** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

L2D_TLB
L2D_TLB_REFILL

Metric group

DTLB_Effectiveness
Other metric group: ITLB_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

l2_tlb_mpki***, L2 Unified TLB MPKI, metric

This metric measures the number of level 2 unified TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L2D_TLB_REFILL / INST_RETIRED * 1000

*** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED
L2D_TLB_REFILL

Metric group

DTLB_Effectiveness
Other metric group: ITLB_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.14 L1I_Cache_Effectiveness metrics for Neoverse V3

L1 Instruction Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L1 Instruction cache on this processor.

Summary of metrics in L1I_Cache_Effectiveness:

- Total metrics: 2

Table 5-14: L1I_Cache_Effectiveness metrics summary

Metric	Name	Description
l1i_cache_miss_ratio	L1I Cache Miss Ratio	This metric measures the ratio of level 1 instruction cache accesses missed to the total number...
l1i_cache_mpki	L1I Cache MPKI	This metric measures the number of level 1 instruction cache accesses missed per thousand...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

l1i_cache_miss_ratio**, L1I Cache Miss Ratio, metric

This metric measures the ratio of level 1 instruction cache accesses missed to the total number of level 1 instruction cache accesses. This gives an indication of the effectiveness of the level 1 instruction cache.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Units

This unit is expressed as per cache access.

Formula

L1I_CACHE_REFILL / L1I_CACHE

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

L1I_CACHE
L1I_CACHE_REFILL

Metric group

L1I_Cache_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l1i_cache_mpki**, L1I Cache MPKI, metric

This metric measures the number of level 1 instruction cache accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

L1I_CACHE_REFILL / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED
L1I_CACHE_REFILL

Metric group

L1I_Cache_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.15 L1D_Cache_Effectiveness metrics for Neoverse V3

L1 Data Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L1 Data Cache on this processor.

Summary of metrics in L1D_Cache_Effectiveness:

- Total metrics: 2

Table 5-15: L1D_Cache_Effectiveness metrics summary

Metric	Name	Description
l1d_cache_miss_ratio	L1D Cache Miss Ratio	This metric measures the ratio of level 1 data cache accesses missed to the total number of level...
l1d_cache_mpki	L1D Cache MPKI	This metric measures the number of level 1 data cache accesses missed per thousand instructions...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

l1d_cache_miss_ratio**, L1D Cache Miss Ratio, metric

This metric measures the ratio of level 1 data cache accesses missed to the total number of level 1 data cache accesses. This gives an indication of the effectiveness of the level 1 data cache.

Units

This unit is expressed as per cache access.

Formula

L1D_CACHE_REFILL / L1D_CACHE

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

L1D_CACHE
L1D_CACHE_REFILL

Metric group

L1D_Cache_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l1d_cache_mpki**, L1D Cache MPKI, metric

This metric measures the number of level 1 data cache accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Formula

L1D_CACHE_REFILL / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED

L1D_CACHE_REFILL

Metric group

L1D_Cache_Effectiveness

Other metric group: MPKI

Methodology

Stage 2

5.16 L2_Cache_Effectiveness metrics for Neoverse V3

L2 Unified Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L2 Unified Cache on this processor.

Summary of metrics in L2_Cache_Effectiveness:

- Total metrics: 2

Table 5-16: L2_Cache_Effectiveness metrics summary

Metric	Name	Description
l2_cache_miss_ratio	L2 Cache Miss Ratio	This metric measures the ratio of level 2 cache accesses missed to the total number of level 2...
l2_cache_mpki	L2 Cache MPKI	This metric measures the number of level 2 unified cache accesses missed per thousand...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

l2_cache_miss_ratio**, L2 Cache Miss Ratio, metric

This metric measures the ratio of level 2 cache accesses missed to the total number of level 2 cache accesses. This gives an indication of the effectiveness of the level 2 cache, which is a unified cache that stores both data and instruction. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a unified cache.

Units

This unit is expressed as per cache access.

Formula

L2D_CACHE_REFILL / L2D_CACHE

** This metric is used in multiple metric groups. See the following for more information.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Events

L2D_CACHE
L2D_CACHE_REFILL

Metric group

L2_Cache_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l2_cache_mpki**, L2 Cache MPKI, metric

This metric measures the number of level 2 unified cache accesses missed per thousand instructions executed. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a unified cache.

Units

This unit is expressed as mpki.

Formula

L2D_CACHE_REFILL / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED
L2D_CACHE_REFILL

Metric group

L2_Cache_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.17 LL_Cache_Effectiveness metrics for Neoverse V3

Last Level Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of Last Level Cache on this processor.

Summary of metrics in LL_Cache_Effectiveness:

- Total metrics: 3

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 5-17: LL_Cache_Effectiveness metrics summary

Metric	Name	Description
ll_cache_read_hit_ratio	LL Cache Read Hit Ratio	This metric measures the ratio of last level cache read accesses hit in the cache to the total...
ll_cache_read_miss_ratio	LL Cache Read Miss Ratio	This metric measures the ratio of last level cache read accesses missed to the total number of...
ll_cache_read_mpki	LL Cache Read MPKI	This metric measures the number of last level cache read accesses missed per thousand...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

ll_cache_read_hit_ratio, LL Cache Read Hit Ratio, metric

This metric measures the ratio of last level cache read accesses hit in the cache to the total number of last level cache accesses. This gives an indication of the effectiveness of the last level cache for read traffic. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a system level cache.

Units

This unit is expressed as per cache access.

Formula

(LL_CACHE_RD - LL_CACHE_MISS_RD) / LL_CACHE_RD

Related telemetry artifacts

Events

LL_CACHE_MISS_RD
LL_CACHE_RD

Metric group

LL_Cache_Effectiveness

Methodology

Stage 2

ll_cache_read_miss_ratio**, LL Cache Read Miss Ratio, metric

This metric measures the ratio of last level cache read accesses missed to the total number of last level cache accesses. This gives an indication of the effectiveness of the last level cache for read traffic. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a system level cache.

Units

This unit is expressed as per cache access.

Formula

LL_CACHE_MISS_RD / LL_CACHE_RD

** This metric is used in multiple metric groups. See the following for more information.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Events

LL_CACHE_MISS_RD
LL_CACHE_RD

Metric group

LL_Cache_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

ll_cache_read_mpki**, LL Cache Read MPKI, metric

This metric measures the number of last level cache read accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

LL_CACHE_MISS_RD / INST_RETIRED * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED
LL_CACHE_MISS_RD

Metric group

LL_Cache_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.18 Operation_Mix metrics for Neoverse V3

Speculative Operation Mix. This metric group provides the distribution of micro-operation types executed for the program.

Summary of metrics in Operation_Mix:

- Total metrics: 9

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 5-18: Operation_Mix metrics summary

Metric	Name	Description
barrier_percentage	Barrier Operations Percentage	This metric measures instruction and data barrier operations as a percentage of operations...
branch_percentage	Branch Operations Percentage	This metric measures branch operations as a percentage of operations speculatively executed.
crypto_percentage	Crypto Operations Percentage	This metric measures crypto operations as a percentage of operations speculatively executed.
integer_dp_percentage	Integer Operations Percentage	This metric measures scalar integer operations as a percentage of operations speculatively executed.
load_percentage	Load Operations Percentage	This metric measures load operations as a percentage of operations speculatively executed.
scalar_fp_percentage	Floating Point Operations Percentage	This metric measures scalar floating point operations as a percentage of operations speculatively...
simd_percentage	Advanced SIMD Operations Percentage	This metric measures advanced SIMD operations as a percentage of total operations speculatively...
store_percentage	Store Operations Percentage	This metric measures store operations as a percentage of operations speculatively executed.
sve_all_percentage	SVE Operations (Load/Store Inclusive) Percentage	This metric measures scalable vector operations, including loads and stores, as a percentage of...

For a complete list of the metrics in Neoverse V3, see Metrics cheat sheet for Neoverse V3 and Metrics lookup table for Neoverse V3.

barrier_percentage, Barrier Operations Percentage, metric

This metric measures instruction and data barrier operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

(ISB_SPEC + DSB_SPEC + DMB_SPEC) / INST_SPEC * 100

Related telemetry artifacts

Events

DMB_SPEC
DSB_SPEC
INST_SPEC
ISB_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

branch_percentage, Branch Operations Percentage, metric

This metric measures branch operations as a percentage of operations speculatively executed.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Units

This unit is expressed as percent of operations.

Formula

(BR_IMMED_SPEC + BR_INDIRECT_SPEC) / INST_SPEC * 100

Related telemetry artifacts

Events

BR_IMMED_SPEC
BR_INDIRECT_SPEC
INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

crypto_percentage, Crypto Operations Percentage, metric

This metric measures crypto operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

CRYPTO_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

CRYPTO_SPEC
INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

integer_dp_percentage, Integer Operations Percentage, metric

This metric measures scalar integer operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

DP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

DP_SPEC

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

load_percentage, Load Operations Percentage, metric

This metric measures load operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

LD_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC
LD_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

scalar_fp_percentage, Floating Point Operations Percentage, metric

This metric measures scalar floating point operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

VFP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC
VFP_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

simd_percentage, Advanced SIMD Operations Percentage, metric

This metric measures advanced SIMD operations as a percentage of total operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

ASE_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

ASE_SPEC
INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

store_percentage, Store Operations Percentage, metric

This metric measures store operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

ST_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC
ST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

sve_all_percentage, SVE Operations (Load/Store Inclusive) Percentage, metric

This metric measures scalable vector operations, including loads and stores, as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

SVE_INST_SPEC / INST_SPEC * 100

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Events

INST_SPEC
SVE_INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6. PMU events by functional group in Neoverse V3

The Performance Monitoring Unit (PMU) collects events through an event interface from other units in the design. These events are used as triggers for event counters. Not all of the possible events are used in the Methodology, however, they are all listed for completeness.

Neoverse V3 provides the following types of PMU events:

- Total implemented Common events: 226
- Total Implemented Product ImpDef events: 27
- PMU Only events : 27
- ETE Only events : 0

PMU events for Neoverse V3 are grouped into the following functional groups:

- Bus, BUS (4 events)
- Chain, CHAIN (1 events)
- Exception, EXCEPTION (15 events)
- L1D_Cache, L1D CACHE (18 events)
- L1I_Cache, L1I CACHE (15 events)
- L2_Cache, L2 CACHE (19 events)
- LL_Cache, LL CACHE (2 events)
- Memory, MEMORY (14 events)
- Retired, RETIRED (24 events)
- SPE, SPE (10 events)
- Spec_Operation, SPEC OPERATION (31 events)
- FP_Operation, FP OPERATION (5 events)
- Stall, STALL (26 events)
- General, GENERAL (7 events)
- TLB, TLB (34 events)
- SVE, SVE (12 events)
- BRBE, BRBE (1 events)
- CPU_Debug, CPU_DEBUG (15 events)

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.1 Bus (BUS) events for Neoverse V3

Bus transaction related events.

Summary of events in Bus:

- Total implemented Common events: 4
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-1: Bus events summary

Code	Mnemonic	Name	Description
0x0019	BUS_ACCESS	Bus access	Counts memory transactions issued by the CPU to the external bus, including snoop requests and...
0x001D	BUS_CYCLES	Bus cycle	Counts bus cycles in the CPU. Bus cycles represent a clock cycle in which a transaction could be...
0x0060	BUS_ACCESS_RD	Bus access, read	Counts memory read transactions seen on the external bus. Each beat of data is counted individually.
0x0061	BUS_ACCESS_WR	Bus access, write	Counts memory write transactions seen on the external bus. Each beat of data is counted...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0019 BUS_ACCESS, Bus access, event

Counts memory transactions issued by the CPU to the external bus, including snoop requests and snoop responses. Each beat of data is counted individually.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Bus

0x001D BUS_CYCLES, Bus cycle, event

Counts bus cycles in the CPU. Bus cycles represent a clock cycle in which a transaction could be sent or received on the interface from the CPU to the external bus. Since that interface is driven at the same clock speed as the CPU, this event is a duplicate of CPU_CYCLES.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Bus

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x0060 BUS_ACCESS_RD, Bus access, read, event

Counts memory read transactions seen on the external bus. Each beat of data is counted individually.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Bus

0x0061 BUS_ACCESS_WR, Bus access, write, event

Counts memory write transactions seen on the external bus. Each beat of data is counted individually.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Bus

6.2 Chain (CHAIN) events for Neoverse V3

Chain related events.

Summary of events in Chain:

- Total implemented Common events: 1
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-2: Chain events summary

Code	Mnemonic	Name	Description
0x001E	CHAIN	Chain a pair of event counters	For odd-numbered counters, this event increments the count by one for each overflow of the...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x001E CHAIN, Chain a pair of event counters, event

For odd-numbered counters, this event increments the count by one for each overflow of the preceding even-numbered counter. For even-numbered counters, there is no increment. This event is used when the even/odd pairs of registers are used as a single counter.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Chain

6.3 Exception (EXCEPTION) events for Neoverse V3

Exception related events.

Summary of events in Exception:

- Total implemented Common events: 15
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-3: Exception events summary

Code	Mnemonic	Name	Description
0x0009	EXC_TAKEN	Exception taken	Counts any taken architecturally visible exceptions such as IRQ, FIQ, SError, and other...
0x000A	EXC_RETURN	Instruction architecturally executed, Condition code check pass, exception return	Counts any architecturally executed exception return instructions. For example: AArch64: ERET
0x0081	EXC_UNDEF	Exception taken, other synchronous	Counts the number of synchronous exceptions which are taken locally that are due to attempting to...
0x0082	EXC_SVC	Exception taken, Supervisor Call	Counts SVC exceptions taken locally.
0x0083	EXC_PABORT	Exception taken, Instruction Abort	Counts synchronous exceptions that are taken locally and caused by Instruction Aborts.
0x0084	EXC_DABORT	Exception taken, Data Abort or SError	Counts exceptions that are taken locally and are caused by data aborts or SErrors. Conditions...
0x0086	EXC_IRQ	Exception taken, IRQ	Counts IRQ exceptions including the virtual IRQs that are taken locally.
0x0087	EXC_FIQ	Exception taken, FIQ	Counts FIQ exceptions including the virtual FIQs that are taken locally.
0x0088	EXC_SMC	Exception taken, Secure Monitor Call	Counts SMC exceptions take to EL3.
0x008A	EXC_HVC	Exception taken, Hypervisor Call	Counts HVC exceptions taken to EL2.
0x008B	EXC_TRAP_PABORT	Exception taken, Instruction Abort not Taken locally	Counts exceptions which are traps not taken locally and are caused by Instruction Aborts. For...
0x008C	EXC_TRAP_DABORT	Exception taken, Data Abort or SError not Taken locally	Counts exceptions which are traps not taken locally and are caused by Data Aborts or SError...
0x008D	EXC_TRAP_OTHER	Exception taken, other traps not Taken locally	Counts the number of synchronous trap exceptions which are not taken locally and are not SVC,...
0x008E	EXC_TRAP_IRQ	Exception taken, IRQ not Taken locally	Counts IRQ exceptions including the virtual IRQs that are not taken locally.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x008F	EXC_TRAP_FIQ	Exception taken, FIQ not Taken locally	Counts FIQs which are not taken locally but taken from EL0, EL1, or EL2 to EL3 (which would be...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0009 EXC_TAKEN, Exception taken, event

Counts any taken architecturally visible exceptions such as IRQ, FIQ, SError, and other synchronous exceptions. Exceptions are counted whether or not they are taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x000A EXC_RETURN, Instruction architecturally executed, Condition code check pass, exception return, event

Counts any architecturally executed exception return instructions. For example: AArch64: ERET

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x0081 EXC_UNDEF, Exception taken, other synchronous, event

Counts the number of synchronous exceptions which are taken locally that are due to attempting to execute an instruction that is **UNDEFINED**. Attempting to execute instruction bit patterns that have not been allocated. Attempting to execute instructions when they are disabled. Attempting to execute instructions at an inappropriate Exception level. Attempting to execute an instruction when the value of PSTATE.IL is 1.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x0082 EXC_SVC, Exception taken, Supervisor Call, event

Counts SVC exceptions taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

Exception

0x0083 EXC_PABORT, Exception taken, Instruction Abort, event

Counts synchronous exceptions that are taken locally and caused by Instruction Aborts.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x0084 EXC_DABORT, Exception taken, Data Abort or SError, event

Counts exceptions that are taken locally and are caused by data aborts or SErrors. Conditions that could cause those exceptions are attempting to read or write memory where the MMU generates a fault, attempting to read or write memory with a misaligned address, interrupts from the nSEI inputs and internally generated SErrors.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x0086 EXC_IRQ, Exception taken, IRQ, event

Counts IRQ exceptions including the virtual IRQs that are taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x0087 EXC_FIQ, Exception taken, FIQ, event

Counts FIQ exceptions including the virtual FIQs that are taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x0088 EXC_SMC, Exception taken, Secure Monitor Call, event

Counts SMC exceptions take to EL3.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x008A EXC_HVC, Exception taken, Hypervisor Call, event

Counts HVC exceptions taken to EL2.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x008B EXC_TRAP_PABORT, Exception taken, Instruction Abort not Taken locally, event

Counts exceptions which are traps not taken locally and are caused by Instruction Aborts. For example, attempting to execute an instruction with a misaligned PC.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x008C EXC_TRAP_DABORT, Exception taken, Data Abort or SError not Taken locally, event

Counts exceptions which are traps not taken locally and are caused by Data Aborts or SError interrupts. Conditions that could cause those exceptions are:

1. Attempting to read or write memory where the MMU generates a fault,
2. Attempting to read or write memory with a misaligned address,
3. Interrupts from the SEI input.
4. internally generated SErrors.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x008D EXC_TRAP_OTHER, Exception taken, other traps not Taken locally, event

Counts the number of synchronous trap exceptions which are not taken locally and are not SVC, SMC, HVC, data aborts, Instruction Aborts, or interrupts.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x008E EXC_TRAP_IRQ, Exception taken, IRQ not Taken locally, event

Counts IRQ exceptions including the virtual IRQs that are not taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

0x008F EXC_TRAP_FIQ, Exception taken, FIQ not Taken locally, event

Counts FIQs which are not taken locally but taken from EL0, EL1, or EL2 to EL3 (which would be the normal behavior for FIQs when not executing in EL3).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Exception

6.4 L1D_Cache (L1D CACHE) events for Neoverse V3

L1 data cache related events.

Summary of events in L1D_Cache:

- Total implemented Common events: 18
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-4: L1D_Cache events summary

Code	Mnemonic	Name	Description
0x0003	L1D_CACHE_REFILL	Level 1 data cache refill	Counts level 1 data cache refills caused by speculatively executed load or store operations that...
0x0004	L1D_CACHE	Level 1 data cache access	Counts level 1 data cache accesses from any load/store operations. Atomic operations that resolve...
0x0015	L1D_CACHE_WB	Level 1 data cache write-back	Counts write-backs of dirty data from the L1 data cache to the L2 cache. This occurs when either...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x0039	L1D_CACHE_LMISS_RD	Level 1 data cache long-latency read miss	Counts cache line refills into the level 1 data cache from any memory read operations, that...
0x0040	L1D_CACHE_RD	Level 1 data cache access, read	Counts level 1 data cache accesses from any load operation. Atomic load operations that resolve...
0x0041	L1D_CACHE_WR	Level 1 data cache access, write	Counts level 1 data cache accesses generated by store operations. This event also counts accesses...
0x0042	L1D_CACHE_REFILL_RD	Level 1 data cache refill, read	Counts level 1 data cache refills caused by speculatively executed load instructions where the...
0x0043	L1D_CACHE_REFILL_WR	Level 1 data cache refill, write	Counts level 1 data cache refills caused by speculatively executed store instructions where the...
0x0044	L1D_CACHE_REFILL_INNER	Level 1 data cache refill, inner	Counts level 1 data cache refills where the cache line data came from caches inside the immediate...
0x0045	L1D_CACHE_REFILL_OUTER	Level 1 data cache refill, outer	Counts level 1 data cache refills for which the cache line data came from outside the immediate...
0x0046	L1D_CACHE_WB_VICTIM	Level 1 data cache write-back, victim	Counts dirty cache line evictions from the level 1 data cache caused by a new cache line...
0x0047	L1D_CACHE_WB_CLEAN	Level 1 data cache write-back, cleaning and coherency	Counts write-backs from the level 1 data cache that are a result of a coherency operation made by...
0x0048	L1D_CACHE_INVAL	Level 1 data cache invalidate	Counts each explicit invalidation of a cache line in the level 1 data cache caused by: • Cache...
0x8140	L1D_CACHE_RW	Level 1 data cache demand access	Counts level 1 data demand cache accesses from any load or store operation. Near atomic...
0x8142	L1D_CACHE_PRFM	Level 1 data cache software preload	Counts level 1 data cache accesses from software preload or prefetch instructions.
0x8144	L1D_CACHE_MISS	Level 1 data cache demand access miss	Counts cache line misses in the level 1 data cache.
0x8146	L1D_CACHE_REFILL_PRFM	Level 1 data cache refill, software preload	Counts level 1 data cache refills where the cache line access was generated by software preload...
0x8154	L1D_CACHE_HWPRF	Level 1 data cache hardware prefetch	Counts level 1 data cache accesses from any load/store operations generated by the hardware...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0003 L1D_CACHE_REFILL, Level 1 data cache refill, event

Counts level 1 data cache refills caused by speculatively executed load or store operations that missed in the level 1 data cache. This event only counts one event per cache line.

Related telemetry artifacts

Metrics

- l1d_cache_mpki in L1D_Cache_Effectiveness
- l1d_cache_mpki in MPKI
- l1d_cache_miss_ratio in L1D_Cache_Effectiveness
- l1d_cache_miss_ratio in Miss_Ratio

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric groups

L1D_Cache_Effectiveness
MPKI
Miss_Ratio

Functional groups

L1D_Cache

0x0004 L1D_CACHE, Level 1 data cache access, event

Counts level 1 data cache accesses from any load/store operations. Atomic operations that resolve in the CPUs caches (near atomic operations) counts as both a write access and read access. Each access to a cache line is counted including the multiple accesses caused by single instructions such as LDM or STM. Each access to other level 1 data or unified memory structures, for example refill buffers, write buffers, and write-back buffers, are also counted.

Related telemetry artifacts

Metrics

- l1d_cache_miss_ratio in L1D_Cache_Effectiveness
- l1d_cache_miss_ratio in Miss_Ratio

Metric groups

L1D_Cache_Effectiveness
Miss_Ratio

Functional groups

L1D_Cache

0x0015 L1D_CACHE_WB, Level 1 data cache write-back, event

Counts write-backs of dirty data from the L1 data cache to the L2 cache. This occurs when either a dirty cache line is evicted from L1 data cache and allocated in the L2 cache or dirty data is written to the L2 and possibly to the next level of cache. This event counts both victim cache line evictions and cache write-backs from snoops or cache maintenance operations. The following cache operations are not counted:

1. Invalidations which do not result in data being transferred out of the L1 (such as evictions of clean data),
2. Full line writes which write to L2 without writing L1, such as write streaming mode.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0039 L1D_CACHE_LMISS_RD, Level 1 data cache long-latency read miss, event

Counts cache line refills into the level 1 data cache from any memory read operations, that incurred additional latency.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0040 L1D_CACHE_RD, Level 1 data cache access, read, event

Counts level 1 data cache accesses from any load operation. Atomic load operations that resolve in the CPUs caches counts as both a write access and read access.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0041 L1D_CACHE_WR, Level 1 data cache access, write, event

Counts level 1 data cache accesses generated by store operations. This event also counts accesses caused by a DC ZVA (data cache zero, specified by virtual address) instruction. Near atomic operations that resolve in the CPUs caches count as a write access and read access.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0042 L1D_CACHE_REFILL_RD, Level 1 data cache refill, read, event

Counts level 1 data cache refills caused by speculatively executed load instructions where the memory read operation misses in the level 1 data cache. This event only counts one event per cache line.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0043 L1D_CACHE_REFILL_WR, Level 1 data cache refill, write, event

Counts level 1 data cache refills caused by speculatively executed store instructions where the memory write operation misses in the level 1 data cache. This event only counts one event per cache line.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

L1D_Cache

0x0044 L1D_CACHE_REFILL_INNER, Level 1 data cache refill, inner, event

Counts level 1 data cache refills where the cache line data came from caches inside the immediate cluster of the core.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0045 L1D_CACHE_REFILL_OUTER, Level 1 data cache refill, outer, event

Counts level 1 data cache refills for which the cache line data came from outside the immediate cluster of the core, like an SLC in the system interconnect or DRAM.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0046 L1D_CACHE_WB_VICTIM, Level 1 data cache write-back, victim, event

Counts dirty cache line evictions from the level 1 data cache caused by a new cache line allocation. This event does not count evictions caused by cache maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0047 L1D_CACHE_WB_CLEAN, Level 1 data cache write-back, cleaning and coherency, event

Counts write-backs from the level 1 data cache that are a result of a coherency operation made by another CPU. Event count includes cache maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

0x0048 L1D_CACHE_INVAL, Level 1 data cache invalidate, event

Counts each explicit invalidation of a cache line in the level 1 data cache caused by:

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Cache Maintenance Operations (CMO) that operate by a virtual address.
- Broadcast cache coherency operations from another CPU in the system.

This event does not count for the following conditions:

1. A cache refill invalidates a cache line.
2. A CMO which is executed on that CPU and invalidates a cache line specified by set/way.

Note that CMOs that operate by set/way cannot be broadcast from one CPU to another.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

`0x8140` L1D_CACHE_RW, Level 1 data cache demand access, event

Counts level 1 data demand cache accesses from any load or store operation. Near atomic operations that resolve in the CPUs caches counts as both a write access and read access.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

`0x8142` L1D_CACHE_PRFM, Level 1 data cache software preload, event

Counts level 1 data cache accesses from software preload or prefetch instructions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

`0x8144` L1D_CACHE_MISS, Level 1 data cache demand access miss, event

Counts cache line misses in the level 1 data cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

`0x8146` L1D_CACHE_REFILL_PRFM, Level 1 data cache refill, software preload, event

Counts level 1 data cache refills where the cache line access was generated by software preload or prefetch instructions.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

`0x8154` L1D_CACHE_HWPRF, Level 1 data cache hardware prefetch, event

Counts level 1 data cache accesses from any load/store operations generated by the hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1D_Cache

6.5 L1I_Cache (L1I CACHE) events for Neoverse V3

L1 instruction cache related events.

Summary of events in L1I_Cache:

- Total implemented Common events: 15
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-5: L1I_Cache events summary

Code	Mnemonic	Name	Description
`0x0001`	L1I_CACHE_REFILL	Level 1 instruction cache refill	Counts cache line refills in the level 1 instruction cache caused by a missed instruction fetch....
`0x0014`	L1I_CACHE	Level 1 instruction cache access	Counts instruction fetches which access the level 1 instruction cache. Instruction cache accesses...
`0x4006`	L1I_CACHE_LMISS	Level 1 instruction cache long-latency miss	Counts cache line refills into the level 1 instruction cache, that incurred additional latency.
`0x8141`	L1I_CACHE_RD	Level 1 instruction cache demand fetch	Counts demand instruction fetches which access the level 1 instruction cache.
`0x8143`	L1I_CACHE_PRFM	Level 1 instruction cache software preload	Counts instruction fetches generated by software preload or prefetch instructions which access...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x8145	L1I_CACHE_HWPRF	Level 1 instruction cache hardware prefetch	Counts instruction fetches which access the level 1 instruction cache generated by the hardware...
0x8147	L1I_CACHE_REFILL_PRFM	Level 1 instruction cache refill, software preload	Counts cache line refills in the level 1 instruction cache caused by a missed instruction fetch...
0x81C0	L1I_CACHE_HIT_RD	Level 1 instruction cache demand fetch hit	Counts demand instruction fetches that access the level 1 instruction cache and hit in the L1...
0x81D0	L1I_CACHE_HIT_RD_FPRFM	Level 1 instruction cache demand fetch first hit, fetched by software preload	Counts demand instruction fetches that access the level 1 instruction cache that hit in the L1...
0x81E0	L1I_CACHE_HIT_RD_FHWPRF	Level 1 instruction cache demand fetch first hit, fetched by hardware prefetcher	Counts demand instruction fetches generated by hardware prefetch that access the level 1...
0x8200	L1I_CACHE_HIT	Level 1 instruction cache hit	Counts instruction fetches that access the level 1 instruction cache and hit in the level 1...
0x8208	L1I_CACHE_HIT_PRFM	Level 1 instruction cache software preload hit	Counts instruction fetches generated by software preload or prefetch instructions that access the...
0x8240	L1I_LFB_HIT_RD	Level 1 instruction cache demand fetch line-fill buffer hit	Counts demand instruction fetches that access the level 1 instruction cache and hit in a line...
0x8250	L1I_LFB_HIT_RD_FPRFM	Level 1 instruction cache demand fetch line-fill buffer first hit, recently fetched by software preload	Counts demand instruction fetches generated by software prefetch instructions that access the...
0x8260	L1I_LFB_HIT_RD_FHWPRF	Level 1 instruction cache demand fetch line-fill buffer first hit, recently fetched by hardware prefetcher	Counts demand instruction fetches generated by hardware prefetch that access the level 1...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0001 L1I_CACHE_REFILL, Level 1 instruction cache refill, event

Counts cache line refills in the level 1 instruction cache caused by a missed instruction fetch. Instruction fetches may include accessing multiple instructions, but the single cache line allocation is counted once.

Related telemetry artifacts

Metrics

- l1i_cache_mpki in L1I_Cache_Effectiveness
- l1i_cache_mpki in MPKI
- l1i_cache_miss_ratio in L1I_Cache_Effectiveness
- l1i_cache_miss_ratio in Miss_Ratio

Metric groups

L1I_Cache_Effectiveness

MPKI

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Miss_Ratio

Functional groups

L1I_Cache

`0x0014` L1I_CACHE, Level 1 instruction cache access, event

Counts instruction fetches which access the level 1 instruction cache. Instruction cache accesses caused by cache maintenance operations are not counted.

Related telemetry artifacts

Metrics

- l1i_cache_miss_ratio in L1I_Cache_Effectiveness
- l1i_cache_miss_ratio in Miss_Ratio

Metric groups

L1I_Cache_Effectiveness
Miss_Ratio

Functional groups

L1I_Cache

`0x4006` L1I_CACHE_LMISS, Level 1 instruction cache long-latency miss, event

Counts cache line refills into the level 1 instruction cache, that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

`0x8141` L1I_CACHE_RD, Level 1 instruction cache demand fetch, event

Counts demand instruction fetches which access the level 1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

`0x8143` L1I_CACHE_PRFM, Level 1 instruction cache software preload, event

Counts instruction fetches generated by software preload or prefetch instructions which access the level 1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

L1I_Cache

0x8145 L1I_CACHE_HWPRF, Level 1 instruction cache hardware prefetch, event

Counts instruction fetches which access the level 1 instruction cache generated by the hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

0x8147 L1I_CACHE_REFILL_PRFM, Level 1 instruction cache refill, software preload, event

Counts cache line refills in the level 1 instruction cache caused by a missed instruction fetch generated by software preload or prefetch instructions. Instruction fetches may include accessing multiple instructions, but the single cache line allocation is counted once.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

0x81C0 L1I_CACHE_HIT_RD, Level 1 instruction cache demand fetch hit, event

Counts demand instruction fetches that access the level 1 instruction cache and hit in the L1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

0x81D0 L1I_CACHE_HIT_RD_FPRFM, Level 1 instruction cache demand fetch first hit, fetched by software preload, event

Counts demand instruction fetches that access the level 1 instruction cache that hit in the L1 instruction cache and the line was requested by a software prefetch.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x81E0 L1I_CACHE_HIT_RD_FHWPRF, Level 1 instruction cache demand fetch first hit, fetched by hardware prefetcher, event

Counts demand instruction fetches generated by hardware prefetch that access the level 1 instruction cache and hit in the L1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

0x8200 L1I_CACHE_HIT, Level 1 instruction cache hit, event

Counts instruction fetches that access the level 1 instruction cache and hit in the level 1 instruction cache. Instruction cache accesses caused by cache maintenance operations are not counted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

0x8208 L1I_CACHE_HIT_PRFM, Level 1 instruction cache software preload hit, event

Counts instruction fetches generated by software preload or prefetch instructions that access the level 1 instruction cache and hit in the level 1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

0x8240 L1I_LFB_HIT_RD, Level 1 instruction cache demand fetch line-fill buffer hit, event

Counts demand instruction fetches that access the level 1 instruction cache and hit in a line that is in the process of being loaded into the level 1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x8250 L1I_LFB_HIT_RD_FPRFM, Level 1 instruction cache demand fetch line-fill buffer first hit, recently fetched by software preload, event

Counts demand instruction fetches generated by software prefetch instructions that access the level 1 instruction cache and hit in a line that is in the process of being loaded into the level 1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

0x8260 L1I_LFB_HIT_RD_FHWPRF, Level 1 instruction cache demand fetch line-fill buffer first hit, recently fetched by hardware prefetcher, event

Counts demand instruction fetches generated by hardware prefetch that access the level 1 instruction cache and hit in a line that is in the process of being loaded into the level 1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L1I_Cache

6.6 L2_Cache (L2 CACHE) events for Neoverse V3

L2 unified cache related events.

Summary of events in L2_Cache:

- Total implemented Common events: 17
- Total Implemented Product ImpDef events: 2
- PMU Only events : 2
- ETE Only events : 0

Table 6-6: L2_Cache events summary

Code	Mnemonic	Name	Description
0x0016	L2D_CACHE	Level 2 data cache access	Counts accesses to the level 2 cache due to data accesses. Level 2 cache is a unified cache for...
0x0017	L2D_CACHE_REFILL	Level 2 data cache refill	Counts cache line refills into the level 2 cache. Level 2 cache is a unified cache for data and...
0x0018	L2D_CACHE_WB	Level 2 data cache write-back	Counts write-backs of data from the L2 cache to outside the CPU. This includes snoops to the L2...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x0050	L2D_CACHE_RD	Level 2 data cache access, read	Counts level 2 data cache accesses due to memory read operations. Level 2 cache is a unified...
0x0051	L2D_CACHE_WR	Level 2 data cache access, write	Counts level 2 cache accesses due to memory write operations. Level 2 cache is a unified cache...
0x0052	L2D_CACHE_REFILL_RD	Level 2 data cache refill, read	Counts refills for memory accesses due to memory read operation counted by L2D_CACHE_RD. Level 2...
0x0053	L2D_CACHE_REFILL_WR	Level 2 data cache refill, write	Counts refills for memory accesses due to memory write operation counted by L2D_CACHE_WR. Level 2...
0x0056	L2D_CACHE_WB_VICTIM	Level 2 data cache write-back, victim	Counts evictions from the level 2 cache because of a line being allocated into the L2 cache.
0x0057	L2D_CACHE_WB_CLEAN	Level 2 data cache write-back, cleaning and coherency	Counts write-backs from the level 2 cache that are a result of either: 1. Cache maintenance...
0x0058	L2D_CACHE_INVAL	Level 2 data cache invalidate	Counts each explicit invalidation of a cache line in the level 2 cache by cache maintenance...
0x01B8	IMP_L2D_CACHE_L1HWPRF	L2D cache access due to L1 hardware prefetch	Counts level 2 cache accesses due to level 1 data cache hardware prefetcher.
0x01B9	IMP_L2D_CACHE_REFILL_L1HWPRF	Refills to L2 cache caused by L1 data cache hardware prefetches	Counts any level 1 hardware prefetch requests that result in a cache line allocation to the L2 cache
0x4009	L2D_CACHE_LMISS_RD	Level 2 data cache long-latency read miss	Counts cache line refills into the level 2 unified cache from any memory read operations that...
0x8148	L2D_CACHE_RW	Level 2 data cache demand access	Counts level 2 cache demand accesses from any load/store operations. Level 2 cache is a unified...
0x814A	L2D_CACHE_PRFM	Level 2 data cache software preload	Counts level 2 data cache accesses generated by software preload or prefetch instructions.
0x814C	L2D_CACHE_MISS	Level 2 data cache demand access miss	Counts cache line misses in the level 2 cache. Level 2 cache is a unified cache for data and...
0x814E	L2D_CACHE_REFILL_PRFM	Level 2 data cache refill, software preload	Counts refills due to accesses generated as a result of software preload or prefetch instructions...
0x8155	L2D_CACHE_HWPRF	Level 2 data cache hardware prefetch	Counts level 2 data cache accesses generated by L2D hardware prefetchers.
0x81BD	L2D_CACHE_REFILL_HWPRF	Level 2 data cache refill, hardware prefetch	Counts level 2 data cache refills where the cache line is requested by a hardware prefetcher.

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0016 L2D_CACHE, Level 2 data cache access, event

Counts accesses to the level 2 cache due to data accesses. Level 2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the first level data cache or translation resolutions due to accesses. This event also counts write back of dirty data from level 1 data cache to the L2 cache.

Related telemetry artifacts

Metrics

- l2_cache_miss_ratio in L2_Cache_Effectiveness

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- l2_cache_miss_ratio in Miss_Ratio

Metric groups

L2_Cache_Effectiveness
Miss_Ratio

Functional groups

L2_Cache

`0x0017` L2D_CACHE_REFILL, Level 2 data cache refill, event

Counts cache line refills into the level 2 cache. Level 2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

Metrics

- l2_cache_mpki in L2_Cache_Effectiveness
- l2_cache_mpki in MPKI
- l2_cache_miss_ratio in L2_Cache_Effectiveness
- l2_cache_miss_ratio in Miss_Ratio

Metric groups

L2_Cache_Effectiveness
MPKI
Miss_Ratio

Functional groups

L2_Cache

`0x0018` L2D_CACHE_WB, Level 2 data cache write-back, event

Counts write-backs of data from the L2 cache to outside the CPU. This includes snoops to the L2 (from other CPUs) which return data even if the snoops cause an invalidation. L2 cache line invalidations which do not write data outside the CPU and snoops which return data from an L1 cache are not counted. Data would not be written outside the cache when invalidating a clean cache line.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

`0x0050` L2D_CACHE_RD, Level 2 data cache access, read, event

Counts level 2 data cache accesses due to memory read operations. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x0051 L2D_CACHE_WR, Level 2 data cache access, write, event

Counts level 2 cache accesses due to memory write operations. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x0052 L2D_CACHE_REFILL_RD, Level 2 data cache refill, read, event

Counts refills for memory accesses due to memory read operation counted by L2D_CACHE_RD. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x0053 L2D_CACHE_REFILL_WR, Level 2 data cache refill, write, event

Counts refills for memory accesses due to memory write operation counted by L2D_CACHE_WR. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x0056 L2D_CACHE_WB_VICTIM, Level 2 data cache write-back, victim, event

Counts evictions from the level 2 cache because of a line being allocated into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

L2_Cache

0x0057 L2D_CACHE_WB_CLEAN, Level 2 data cache write-back, cleaning and coherency, event

Counts write-backs from the level 2 cache that are a result of either:

1. Cache maintenance operations,
2. Snoop responses or,
3. Direct cache transfers to another CPU due to a forwarding snoop request.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x0058 L2D_CACHE_INVAL, Level 2 data cache invalidate, event

Counts each explicit invalidation of a cache line in the level 2 cache by cache maintenance operations that operate by a virtual address, or by external coherency operations. This event does not count if either:

1. A cache refill invalidates a cache line or,
2. A Cache Maintenance Operation (CMO), which invalidates a cache line specified by set/way, is executed on that CPU.

CMOs that operate by set/way cannot be broadcast from one CPU to another.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x01B8 IMP_L2D_CACHE_L1HWPRF, L2D cache access due to L1 hardware prefetch, event

Counts level 2 cache accesses due to level 1 data cache hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x01B9 IMP_L2D_CACHE_REFILL_L1HWPRF, Refills to L2 cache caused by L1 data cache hardware prefetches, event

Counts any level 1 hardware prefetch requests that result in a cache line allocation to the L2 cache

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x4009 L2D_CACHE_LMISS_RD, Level 2 data cache long-latency read miss, event

Counts cache line refills into the level 2 unified cache from any memory read operations that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x8148 L2D_CACHE_RW, Level 2 data cache demand access, event

Counts level 2 cache demand accesses from any load/store operations. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x814A L2D_CACHE_PRFM, Level 2 data cache software preload, event

Counts level 2 data cache accesses generated by software preload or prefetch instructions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x814C L2D_CACHE_MISS, Level 2 data cache demand access miss, event

Counts cache line misses in the level 2 cache. Level 2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x814E L2D_CACHE_REFILL_PRFM, Level 2 data cache refill, software preload, event

Counts refills due to accesses generated as a result of software preload or prefetch instructions as counted by L2D_CACHE_PRFM.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x8155 L2D_CACHE_HWPRF, Level 2 data cache hardware prefetch, event

Counts level 2 data cache accesses generated by L2D hardware prefetchers.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

0x81BD L2D_CACHE_REFILL_HWPRF, Level 2 data cache refill, hardware prefetch, event

Counts level 2 data cache refills where the cache line is requested by a hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

L2_Cache

6.7 LL_Cache (LL CACHE) events for Neoverse V3

Last Level Cache related events.

Summary of events in LL_Cache:

- Total implemented Common events: 2
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 6-7: LL_Cache events summary

Code	Mnemonic	Name	Description
0x0036	LL_CACHE_RD	Last level cache access, read	Counts read transactions that were returned from outside the core cluster. This event counts for...
0x0037	LL_CACHE_MISS_RD	Last level cache miss, read	Counts read transactions that were returned from outside the core cluster but missed in the...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0036 LL_CACHE_RD, Last level cache access, read, event

Counts read transactions that were returned from outside the core cluster. This event counts for external last level cache when the system register CPUECTLR.EXTLLC bit is set. This event counts read transactions returned from outside the core if those transactions are either hit in the system level cache or missed in the SLC and are returned from any other external sources.

Related telemetry artifacts

Metrics

- ll_cache_read_miss_ratio in LL_Cache_Effectiveness
- ll_cache_read_miss_ratio in Miss_Ratio
- ll_cache_read_hit_ratio

Metric groups

LL_Cache_Effectiveness

Miss_Ratio

Functional groups

LL_Cache

0x0037 LL_CACHE_MISS_RD, Last level cache miss, read, event

Counts read transactions that were returned from outside the core cluster but missed in the system level cache. This event counts for external last level cache when the system register CPUECTLR.EXTLLC bit is set. This event counts read transactions returned from outside the core if those transactions are missed in the System level Cache. The data source of the transaction is indicated by a field in the CHI transaction returning to the CPU. This event does not count reads caused by cache maintenance operations.

Related telemetry artifacts

Metrics

- ll_cache_read_mpki in LL_Cache_Effectiveness
- ll_cache_read_mpki in MPKI
- ll_cache_read_miss_ratio in LL_Cache_Effectiveness
- ll_cache_read_miss_ratio in Miss_Ratio
- ll_cache_read_hit_ratio

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric groups

LL_Cache_Effectiveness
MPKI
Miss_Ratio

Functional groups

LL_Cache

6.8 Memory (MEMORY) events for Neoverse V3

Memory system related events.

Summary of events in Memory:

- Total implemented Common events: 14
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-8: Memory events summary

Code	Mnemonic	Name	Description
0x0013	MEM_ACCESS	Data memory access	Counts memory accesses issued by the CPU load store unit, where those accesses are issued due to...
0x001A	MEMORY_ERROR	Local memory error	Counts any detected correctable or uncorrectable physical memory errors (ECC or parity) in...
0x0031	REMOTE_ACCESS	Access to another socket in a multi-socket system	Counts accesses to another chip, which is implemented as a different CMN mesh in the system. If...
0x0066	MEM_ACCESS_RD	Data memory access, read	Counts memory accesses issued by the CPU due to load operations. The event counts any memory load...
0x0067	MEM_ACCESS_WR	Data memory access, write	Counts memory accesses issued by the CPU due to store operations. The event counts any memory...
0x4020	LDST_ALIGN_LAT	Access with additional latency from alignment	Counts the number of memory read and write accesses in a cycle that incurred additional latency,...
0x4021	LD_ALIGN_LAT	Load with additional latency from alignment	Counts the number of memory read accesses in a cycle that incurred additional latency, due to the...
0x4022	ST_ALIGN_LAT	Store with additional latency from alignment	Counts the number of memory write access in a cycle that incurred additional latency, due to the...
0x4024	MEM_ACCESS_CHECKED	Checked data memory access	Counts the number of memory read and write accesses counted by MEM_ACCESS that are tag checked by...
0x4025	MEM_ACCESS_CHECKED_RD	Checked data memory access, read	Counts the number of memory read accesses in a cycle that are tag checked by the Memory Tagging...
0x4026	MEM_ACCESS_CHECKED_WR	Checked data memory access, write	Counts the number of memory write accesses in a cycle that is tag checked by the Memory Tagging...
0x8120	INST_FETCH_PERCYC	Event in progress, INST FETCH	Counts number of instruction fetches outstanding per cycle, which will provide an average latency...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x8121	MEM_ACCESS_RD_PERCYC	Event in progress, MEM ACCESS RD	Counts the number of outstanding loads or memory read accesses per cycle.
0x8124	INST_FETCH	Instruction memory access	Counts Instruction memory accesses that the PE makes.

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0013 MEM_ACCESS, Data memory access, event

Counts memory accesses issued by the CPU load store unit, where those accesses are issued due to load or store operations. This event counts memory accesses no matter whether the data is received from any level of cache hierarchy or external memory. If memory accesses are broken up into smaller transactions than what were specified in the load or store instructions, then the event counts those smaller memory transactions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x001A MEMORY_ERROR, Local memory error, event

Counts any detected correctable or uncorrectable physical memory errors (ECC or parity) in protected CPUs RAMs. On the core, this event counts errors in the caches (including data and tag rams). Any detected memory error (from either a speculative and abandoned access, or an architecturally executed access) is counted. Note that errors are only detected when the actual protected memory is accessed by an operation.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x0031 REMOTE_ACCESS, Access to another socket in a multi-socket system, event

Counts accesses to another chip, which is implemented as a different CMN mesh in the system. If the CHI bus response back to the core indicates that the data source is from another chip (mesh), then the counter is updated. If no data is returned, even if the system snoops another chip/mesh, then the counter is not updated.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x0066 MEM_ACCESS_RD, Data memory access, read, event

Counts memory accesses issued by the CPU due to load operations. The event counts any memory load access, no matter whether the data is received from any level of cache hierarchy or external memory. The event also counts atomic load operations. If memory accesses are broken up by the load/store unit into smaller transactions that are issued by the bus interface, then the event counts those smaller transactions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x0067 MEM_ACCESS_WR, Data memory access, write, event

Counts memory accesses issued by the CPU due to store operations. The event counts any memory store access, no matter whether the data is located in any level of cache or external memory. The event also counts atomic load and store operations. If memory accesses are broken up by the load/store unit into smaller transactions that are issued by the bus interface, then the event counts those smaller transactions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x4020 LDST_ALIGN_LAT, Access with additional latency from alignment, event

Counts the number of memory read and write accesses in a cycle that incurred additional latency, due to the alignment of the address and the size of data being accessed, which results in store crossing a single cache line.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x4021 LD_ALIGN_LAT, Load with additional latency from alignment, event

Counts the number of memory read accesses in a cycle that incurred additional latency, due to the alignment of the address and size of data being accessed, which results in load crossing a single cache line.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

Memory

0x4022 ST_ALIGN_LAT, Store with additional latency from alignment, event

Counts the number of memory write access in a cycle that incurred additional latency, due to the alignment of the address and size of data being accessed incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x4024 MEM_ACCESS_CHECKED, Checked data memory access, event

Counts the number of memory read and write accesses counted by MEM_ACCESS that are tag checked by the Memory Tagging Extension (MTE). This event is implemented as the sum of MEM_ACCESS_CHECKED_RD and MEM_ACCESS_CHECKED_WR

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x4025 MEM_ACCESS_CHECKED_RD, Checked data memory access, read, event

Counts the number of memory read accesses in a cycle that are tag checked by the Memory Tagging Extension (MTE).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x4026 MEM_ACCESS_CHECKED_WR, Checked data memory access, write, event

Counts the number of memory write accesses in a cycle that is tag checked by the Memory Tagging Extension (MTE).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x8120 INST_FETCH_PERCYC, Event in progress, INST FETCH, event

Counts number of instruction fetches outstanding per cycle, which will provide an average latency of instruction fetch.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x8121 MEM_ACCESS_RD_PERCYC, Event in progress, MEM ACCESS RD, event

Counts the number of outstanding loads or memory read accesses per cycle.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

0x8124 INST_FETCH, Instruction memory access, event

Counts Instruction memory accesses that the PE makes.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Memory

6.9 Retired (RETIRED) events for Neoverse V3

Retired instruction and operation events.

Summary of events in Retired:

- Total implemented Common events: 24
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-9: Retired events summary

Code	Mnemonic	Name	Description
0x0000	SW_INCR	Instruction architecturally executed, Condition code check pass, software increment	Counts software writes to the PMSWINC_EL0 (software PMU increment) register. The PMSWINC_EL0...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x0008	INST_RETIRED	Instruction architecturally executed	Counts instructions that have been architecturally executed.
0x000B	CID_WRITE_RETIRED	Instruction architecturally executed, Condition code check pass, write to CONTEXTIDR	Counts architecturally executed writes to the CONTEXTIDR_EL1 register, which usually contain the...
0x000D	BR_IMMED_RETIRED	Branch instruction architecturally executed, immediate	Counts architecturally executed direct branches.
0x000E	BR_RETURN_RETIRED	Branch instruction architecturally executed, procedure return, taken	Counts architecturally executed procedure returns.
0x001C	TTBR_WRITE_RETIRED	Instruction architecturally executed, Condition code check pass, write to TTBR	Counts architectural writes to TTBR0/1_EL1. If virtualization host extensions are enabled (by...
0x0021	BR_RETIRED	Instruction architecturally executed, branch	Counts architecturally executed branches, whether the branch is taken or not. Instructions that...
0x0022	BR_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted	Counts branches counted by BR_RETIRED which were mispredicted and caused a pipeline flush.
0x003A	OP_RETIRED	Micro-operation architecturally executed	Counts micro-operations that are architecturally executed. This is a count of number of...
0x810C	BR_INDNR_TAKEN_RETIRED	Branch instruction architecturally executed, indirect excluding procedure return, taken	Counts architecturally executed indirect branches excluding procedure returns that were taken.
0x8110	BR_IMMED_PRED_RETIRED	Branch instruction architecturally executed, predicted immediate	Counts architecturally executed direct branches that were correctly predicted.
0x8111	BR_IMMED_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted immediate	Counts architecturally executed direct branches that were mispredicted and caused a pipeline flush.
0x8112	BR_IND_PRED_RETIRED	Branch instruction architecturally executed, predicted indirect	Counts architecturally executed indirect branches including procedure returns that were correctly...
0x8113	BR_IND_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted indirect	Counts architecturally executed indirect branches including procedure returns that were...
0x8114	BR_RETURN_PRED_RETIRED	Branch instruction architecturally executed, predicted procedure return	Counts architecturally executed procedure returns that were correctly predicted.
0x8115	BR_RETURN_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted procedure return	Counts architecturally executed procedure returns that were mispredicted and caused a pipeline...
0x8116	BR_INDNR_PRED_RETIRED	Branch instruction architecturally executed, predicted indirect excluding procedure return	Counts architecturally executed indirect branches excluding procedure returns that were correctly...
0x8117	BR_INDNR_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted indirect excluding procedure return	Counts architecturally executed indirect branches excluding procedure returns that were...
0x8118	BR_TAKEN_PRED_RETIRED	Branch instruction architecturally executed, predicted branch, taken	Counts architecturally executed branches that were taken and were correctly predicted.
0x8119	BR_TAKEN_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted branch, taken	Counts architecturally executed branches that were taken and were mispredicted causing a pipeline...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x811A	BR_SKIP_PRED_RETIRED	Branch instruction architecturally executed, predicted branch, not taken	Counts architecturally executed branches that were not taken and were correctly predicted.
0x811B	BR_SKIP_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted branch, not taken	Counts architecturally executed branches that were not taken and were mispredicted causing a...
0x811C	BR_PRED_RETIRED	Branch instruction architecturally executed, predicted branch	Counts branch instructions counted by BR_RETIRED which were correctly predicted.
0x811D	BR_IND_RETIRED	Instruction architecturally executed, indirect branch	Counts architecturally executed indirect branches including procedure returns.

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0000 SW_INCR, Instruction architecturally executed, Condition code check pass, software increment, event

Counts software writes to the PMSWINC_EL0 (software PMU increment) register. The PMSWINC_EL0 register is a manually updated counter for use by application software.

This event could be used to measure any user program event, such as accesses to a particular data structure (by writing to the PMSWINC_EL0 register each time the data structure is accessed).

To use the PMSWINC_EL0 register and event, developers must insert instructions that write to the PMSWINC_EL0 register into the source code.

Since the SW_INCR event records writes to the PMSWINC_EL0 register, there is no need to do a read/increment/write sequence to the PMSWINC_EL0 register.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x0008 INST_RETIRED, Instruction architecturally executed, event

Counts instructions that have been architecturally executed.

Related telemetry artifacts

Metrics

- ipc
- branch_mpki in Branch_Effectiveness
- branch_mpki in MPKI
- itlb_mpki in ITLB_Effectiveness
- itlb_mpki in MPKI
- l1i_tlb_mpki in ITLB_Effectiveness

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- l1i_tlb_mpki in MPKI
- dtlb_mpki in DTLB_Effectiveness
- dtlb_mpki in MPKI
- l1d_tlb_mpki in DTLB_Effectiveness
- l1d_tlb_mpki in MPKI
- l2_tlb_mpki in DTLB_Effectiveness
- l2_tlb_mpki in ITLB_Effectiveness
- l2_tlb_mpki in MPKI
- l1i_cache_mpki in L1I_Cache_Effectiveness
- l1i_cache_mpki in MPKI
- l1d_cache_mpki in L1D_Cache_Effectiveness
- l1d_cache_mpki in MPKI
- l2_cache_mpki in L2_Cache_Effectiveness
- l2_cache_mpki in MPKI
- ll_cache_read_mpki in LL_Cache_Effectiveness
- ll_cache_read_mpki in MPKI

Metric groups

Branch_Effectiveness
DTLB_Effectiveness
General
ITLB_Effectiveness
L1D_Cache_Effectiveness
L1I_Cache_Effectiveness
L2_Cache_Effectiveness
LL_Cache_Effectiveness
MPKI

Functional groups

Retired

0x000B CID_WRITE_RETIRED, Instruction architecturally executed, Condition code check pass, write to CONTEXTIDR, event

Counts architecturally executed writes to the CONTEXTIDR_EL1 register, which usually contain the kernel PID and can be output with hardware trace.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x000D BR_IMMED_RETIRED, Branch instruction architecturally executed, immediate, event

Counts architecturally executed direct branches.

Related telemetry artifacts

Metrics

- branch_direct_ratio

Metric groups

Branch_Effectiveness

Functional groups

Retired

0x000E BR_RETURN_RETIRED, Branch instruction architecturally executed, procedure return, taken, event

Counts architecturally executed procedure returns.

Related telemetry artifacts

Metrics

- branch_return_ratio

Metric groups

Branch_Effectiveness

Functional groups

Retired

0x001C TTBR_WRITE_RETIRED, Instruction architecturally executed, Condition code check pass, write to TTBR, event

Counts architectural writes to TTBR0/1_EL1. If virtualization host extensions are enabled (by setting the HCR_EL2.E2H bit to 1), then accesses to TTBR0/1_EL1 that are redirected to TTBR0/1_EL2, or accesses to TTBR0/1_EL12, are counted. TTBRn registers are typically updated when the kernel is swapping user-space threads or applications.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x0021 BR_RETIRED, Instruction architecturally executed, branch, event

Counts architecturally executed branches, whether the branch is taken or not. Instructions that explicitly write to the PC are also counted. Note that exception generating instructions, exception return instructions and context synchronization instructions are not counted.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Metrics

- branch_direct_ratio
- branch_indirect_ratio
- branch_return_ratio
- branch_misprediction_ratio in Branch_Effectiveness
- branch_misprediction_ratio in Miss_Ratio

Metric groups

Branch_Effectiveness
Miss_Ratio

Functional groups

Retired

0x0022 BR_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted, event

Counts branches counted by BR_RETIRED which were mispredicted and caused a pipeline flush.

Related telemetry artifacts

Metrics

- branch_mpki in Branch_Effectiveness
- branch_mpki in MPKI
- branch_misprediction_ratio in Branch_Effectiveness
- branch_misprediction_ratio in Miss_Ratio

Metric groups

Branch_Effectiveness
MPKI
Miss_Ratio

Functional groups

Retired

0x003A OP_RETIRED, Micro-operation architecturally executed, event

Counts micro-operations that are architecturally executed. This is a count of number of micro-operations retired from the commit queue in a single cycle.

Related telemetry artifacts

Metrics

- retiring
- bad_speculation

Metric groups

Topdown_L1

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

Retired

0x810C BR_INDNR_TAKEN_RETIRED, Branch instruction architecturally executed, indirect excluding procedure return, taken, event

Counts architecturally executed indirect branches excluding procedure returns that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x8110 BR_IMMED_PRED_RETIRED, Branch instruction architecturally executed, predicted immediate, event

Counts architecturally executed direct branches that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x8111 BR_IMMED_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted immediate, event

Counts architecturally executed direct branches that were mispredicted and caused a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x8112 BR_IND_PRED_RETIRED, Branch instruction architecturally executed, predicted indirect, event

Counts architecturally executed indirect branches including procedure returns that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

`0x8113` BR_IND_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted indirect, event

Counts architecturally executed indirect branches including procedure returns that were mispredicted and caused a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

`0x8114` BR_RETURN_PRED_RETIRED, Branch instruction architecturally executed, predicted procedure return, event

Counts architecturally executed procedure returns that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

`0x8115` BR_RETURN_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted procedure return, event

Counts architecturally executed procedure returns that were mispredicted and caused a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

`0x8116` BR_INDNR_PRED_RETIRED, Branch instruction architecturally executed, predicted indirect excluding procedure return, event

Counts architecturally executed indirect branches excluding procedure returns that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x8117 BR_INDNR_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted indirect excluding procedure return, event

Counts architecturally executed indirect branches excluding procedure returns that were mispredicted and caused a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x8118 BR_TAKEN_PRED_RETIRED, Branch instruction architecturally executed, predicted branch, taken, event

Counts architecturally executed branches that were taken and were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x8119 BR_TAKEN_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted branch, taken, event

Counts architecturally executed branches that were taken and were mispredicted causing a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x811A BR_SKIP_PRED_RETIRED, Branch instruction architecturally executed, predicted branch, not taken, event

Counts architecturally executed branches that were not taken and were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x811B BR_SKIP_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted branch, not taken, event

Counts architecturally executed branches that were not taken and were mispredicted causing a pipeline flush.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x811C BR_PRED_RETIRED, Branch instruction architecturally executed, predicted branch, event

Counts branch instructions counted by BR_RETIRED which were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Retired

0x811D BR_IND_RETIRED, Instruction architecturally executed, indirect branch, event

Counts architecturally executed indirect branches including procedure returns.

Related telemetry artifacts

Metrics

- branch_indirect_ratio

Metric groups

Branch_Effectiveness

Functional groups

Retired

6.10 SPE (SPE) events for Neoverse V3

SPE related events.

Summary of events in SPE:

- Total implemented Common events: 10
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-10: SPE events summary

Code	Mnemonic	Name	Description
0x4000	SAMPLE_POP	Statistical Profiling sample population	Counts statistical profiling sample population, the count of all operations that could be sampled...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x4001	SAMPLE_FEED	Statistical Profiling sample taken	Counts statistical profiling samples taken for sampling.
0x4002	SAMPLE_FILTRATE	Statistical Profiling sample taken and not removed by filtering	Counts statistical profiling samples taken which are not removed by filtering.
0x4003	SAMPLE_COLLISION	Statistical Profiling sample collided with previous sample	Counts statistical profiling samples that have collided with a previous sample and so therefore...
0x812A	SAMPLE_FEED_BR	Statistical Profiling sample taken, branch	Counts statistical profiling samples taken which are branches.
0x812B	SAMPLE_FEED_LD	Statistical Profiling sample taken, load	Counts statistical profiling samples taken which are loads or load atomic operations.
0x812C	SAMPLE_FEED_ST	Statistical Profiling sample taken, store	Counts statistical profiling samples taken which are stores or store atomic operations.
0x812D	SAMPLE_FEED_OP	Statistical Profiling sample taken, matching operation type	Counts statistical profiling samples taken which are matching any operation type filters supported.
0x812E	SAMPLE_FEED_EVENT	Statistical Profiling sample taken, matching events	Counts statistical profiling samples taken which are matching event packet filter constraints.
0x812F	SAMPLE_FEED_LAT	Statistical Profiling sample taken, exceeding minimum latency	Counts statistical profiling samples taken which are exceeding minimum latency set by operation...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x4000 SAMPLE_POP, Statistical Profiling sample population, event

Counts statistical profiling sample population, the count of all operations that could be sampled but may or may not be chosen for sampling.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

0x4001 SAMPLE_FEED, Statistical Profiling sample taken, event

Counts statistical profiling samples taken for sampling.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

0x4002 SAMPLE_FILTRATE, Statistical Profiling sample taken and not removed by filtering, event

Counts statistical profiling samples taken which are not removed by filtering.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

0x4003 SAMPLE_COLLISION, Statistical Profiling sample collided with previous sample, event

Counts statistical profiling samples that have collided with a previous sample and so therefore not taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

0x812A SAMPLE_FEED_BR, Statistical Profiling sample taken, branch, event

Counts statistical profiling samples taken which are branches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

0x812B SAMPLE_FEED_LD, Statistical Profiling sample taken, load, event

Counts statistical profiling samples taken which are loads or load atomic operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

0x812C SAMPLE_FEED_ST, Statistical Profiling sample taken, store, event

Counts statistical profiling samples taken which are stores or store atomic operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x812D SAMPLE_FEED_OP, Statistical Profiling sample taken, matching operation type, event

Counts statistical profiling samples taken which are matching any operation type filters supported.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

0x812E SAMPLE_FEED_EVENT, Statistical Profiling sample taken, matching events, event

Counts statistical profiling samples taken which are matching event packet filter constraints.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

0x812F SAMPLE_FEED_LAT, Statistical Profiling sample taken, exceeding minimum latency, event

Counts statistical profiling samples taken which are exceeding minimum latency set by operation latency filter constraints.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SPE

6.11 Spec_Operation (SPEC OPERATION) events for Neoverse V3

Speculatively executed operations related events.

Summary of events in Spec_Operation:

- Total implemented Common events: 31
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 6-11: Spec_Operation events summary

Code	Mnemonic	Name	Description
0x0010	BR_MIS_PRED	Branch instruction speculatively executed, mispredicted or not predicted	Counts branches which are speculatively executed and mispredicted.
0x0012	BR_PRED	Predictable branch instruction speculatively executed	Counts all speculatively executed branches.
0x001B	INST_SPEC	Operation speculatively executed	Counts operations that have been speculatively executed.
0x003B	OP_SPEC	Micro-operation speculatively executed	Counts micro-operations speculatively executed. This is the count of the number of...
0x0068	UNALIGNED_LD_SPEC	Unaligned access, read	Counts unaligned memory read operations issued by the CPU. This event counts unaligned accesses...
0x0069	UNALIGNED_ST_SPEC	Unaligned access, write	Counts unaligned memory write operations issued by the CPU. This event counts unaligned accesses...
0x006A	UNALIGNED_LDST_SPEC	Unaligned access	Counts unaligned memory operations issued by the CPU. This event counts unaligned accesses (as...
0x006C	LDREX_SPEC	Exclusive operation speculatively executed, Load-Exclusive	Counts Load-Exclusive operations that have been speculatively executed. For example: LDREX, LDX
0x006D	STREX_PASS_SPEC	Exclusive operation speculatively executed, Store-Exclusive pass	Counts store-exclusive operations that have been speculatively executed and have successfully...
0x006E	STREX_FAIL_SPEC	Exclusive operation speculatively executed, Store-Exclusive fail	Counts store-exclusive operations that have been speculatively executed and have not successfully...
0x006F	STREX_SPEC	Exclusive operation speculatively executed, Store-Exclusive	Counts store-exclusive operations that have been speculatively executed.
0x0070	LD_SPEC	Operation speculatively executed, load	Counts speculatively executed load operations including Single Instruction Multiple Data (SIMD)...
0x0071	ST_SPEC	Operation speculatively executed, store	Counts speculatively executed store operations including Single Instruction Multiple Data (SIMD)...
0x0072	LDST_SPEC	Operation speculatively executed, load or store	Counts load and store operations that have been speculatively executed.
0x0073	DP_SPEC	Operation speculatively executed, integer data processing	Counts speculatively executed logical or arithmetic instructions such as MOV/MVN operations.
0x0074	ASE_SPEC	Operation speculatively executed, Advanced SIMD	Counts speculatively executed Advanced SIMD operations excluding load, store and move...
0x0075	VFP_SPEC	Operation speculatively executed, scalar floating-point	Counts speculatively executed floating point operations. This event does not count operations...
0x0076	PC_WRITE_SPEC	Operation speculatively executed, Software change of the PC	Counts speculatively executed operations which cause software changes of the PC. Those operations...
0x0077	CRYPTO_SPEC	Operation speculatively executed, Cryptographic instruction	Counts speculatively executed cryptographic operations except for PMULL and VMULL operations.
0x0078	BR_IMMED_SPEC	Branch speculatively executed, immediate branch	Counts direct branch operations which are speculatively executed.
0x0079	BR_RETURN_SPEC	Branch speculatively executed, procedure return	Counts procedure return operations (RET, RETAA and RETAB) which are speculatively executed.
0x007A	BR_INDIRECT_SPEC	Branch speculatively executed, indirect branch	Counts indirect branch operations including procedure returns, which are speculatively executed....
0x007C	ISB_SPEC	Barrier speculatively executed, ISB	Counts ISB operations that are executed.
0x007D	DSB_SPEC	Barrier speculatively executed, DSB	Counts DSB operations that are speculatively issued to Load/Store unit in the CPU.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x007E	DMB_SPEC	Barrier speculatively executed, DMB	Counts DMB operations that are speculatively issued to the Load/Store unit in the CPU. This event...
0x0090	RC_LD_SPEC	Release consistency operation speculatively executed, Load-Acquire	Counts any load acquire operations that are speculatively executed. For example: LDAR, LDARH, LDARB
0x0091	RC_ST_SPEC	Release consistency operation speculatively executed, Store-Release	Counts any store release operations that are speculatively executed. For example: STLR, STLRH, STLRB
0x8004	SIMD_INST_SPEC	Operation speculatively executed, SIMD	Counts speculatively executed operations that are SIMD or SVE vector operations or Advanced SIMD...
0x8005	ASE_INST_SPEC	Operation speculatively executed, Advanced SIMD	Counts speculatively executed Advanced SIMD operations.
0x8040	INT_SPEC	Integer operation speculatively executed	Counts speculatively executed integer arithmetic operations.
0x8087	PRF_SPEC	Operation speculatively executed, Prefetch	Counts speculatively executed operations that prefetch memory. For example: Scalar: PRFM, SVE:...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0010 BR_MIS_PRED, Branch instruction speculatively executed, mispredicted or not predicted, event

Counts branches which are speculatively executed and mispredicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x0012 BR_PRED, Predictable branch instruction speculatively executed, event

Counts all speculatively executed branches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x001B INST_SPEC, Operation speculatively executed, event

Counts operations that have been speculatively executed.

Related telemetry artifacts

Metrics

- load_percentage
- store_percentage
- integer_dp_percentage

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- simd_percentage
- scalar_fp_percentage
- barrier_percentage
- branch_percentage
- crypto_percentage
- sve_all_percentage
- sve_predicate_percentage
- fp16_percentage
- fp32_percentage
- fp64_percentage

Metric groups

FP_Precision_Mix
Operation_Mix
SVE_Effectiveness

Functional groups

Spec_Operation

0x003B OP_SPEC, Micro-operation speculatively executed, event

Counts micro-operations speculatively executed. This is the count of the number of micro-operations dispatched in a cycle.

Related telemetry artifacts

Metrics

- retiring
- bad_speculation

Metric groups

Topdown_L1

Functional groups

Spec_Operation

0x0068 UNALIGNED_LD_SPEC, Unaligned access, read, event

Counts unaligned memory read operations issued by the CPU. This event counts unaligned accesses (as defined by the actual instruction), even if they are subsequently issued as multiple aligned accesses. The event does not count preload operations (PLD, PLI).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x0069 UNALIGNED_ST_SPEC, Unaligned access, write, event

Counts unaligned memory write operations issued by the CPU. This event counts unaligned accesses (as defined by the actual instruction), even if they are subsequently issued as multiple aligned accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x006A UNALIGNED_LDST_SPEC, Unaligned access, event

Counts unaligned memory operations issued by the CPU. This event counts unaligned accesses (as defined by the actual instruction), even if they are subsequently issued as multiple aligned accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x006C LDREX_SPEC, Exclusive operation speculatively executed, Load-Exclusive, event

Counts Load-Exclusive operations that have been speculatively executed. For example: LDREX, LDX

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x006D STREX_PASS_SPEC, Exclusive operation speculatively executed, Store-Exclusive pass, event

Counts store-exclusive operations that have been speculatively executed and have successfully completed the store operation.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x006E STREX_FAIL_SPEC, Exclusive operation speculatively executed, Store-Exclusive fail, event

Counts store-exclusive operations that have been speculatively executed and have not successfully completed the store operation.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x006F STREX_SPEC, Exclusive operation speculatively executed, Store-Exclusive, event

Counts store-exclusive operations that have been speculatively executed.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x0070 LD_SPEC, Operation speculatively executed, load, event

Counts speculatively executed load operations including Single Instruction Multiple Data (SIMD) load operations.

Related telemetry artifacts

Metrics

- load_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x0071 ST_SPEC, Operation speculatively executed, store, event

Counts speculatively executed store operations including Single Instruction Multiple Data (SIMD) store operations.

Related telemetry artifacts

Metrics

- store_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x0072 LDST_SPEC, Operation speculatively executed, load or store, event

Counts load and store operations that have been speculatively executed.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

`0x0073` DP_SPEC, Operation speculatively executed, integer data processing, event

Counts speculatively executed logical or arithmetic instructions such as MOV/MVN operations.

Related telemetry artifacts

Metrics

- integer_dp_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

`0x0074` ASE_SPEC, Operation speculatively executed, Advanced SIMD, event

Counts speculatively executed Advanced SIMD operations excluding load, store and move micro-operations that move data to or from SIMD (vector) registers.

Related telemetry artifacts

Metrics

- simd_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

`0x0075` VFP_SPEC, Operation speculatively executed, scalar floating-point, event

Counts speculatively executed floating point operations. This event does not count operations that move data to or from floating point (vector) registers.

Related telemetry artifacts

Metrics

- scalar_fp_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x0076 PC_WRITE_SPEC, Operation speculatively executed, Software change of the PC, event

Counts speculatively executed operations which cause software changes of the PC. Those operations include all taken branch operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x0077 CRYPTO_SPEC, Operation speculatively executed, Cryptographic instruction, event

Counts speculatively executed cryptographic operations except for PMULL and VMULL operations.

Related telemetry artifacts

Metrics

- crypto_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x0078 BR_IMMED_SPEC, Branch speculatively executed, immediate branch, event

Counts direct branch operations which are speculatively executed.

Related telemetry artifacts

Metrics

- branch_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x0079 BR_RETURN_SPEC, Branch speculatively executed, procedure return, event

Counts procedure return operations (RET, RETAA and RETAB) which are speculatively executed.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x007A BR_INDIRECT_SPEC, Branch speculatively executed, indirect branch, event

Counts indirect branch operations including procedure returns, which are speculatively executed. This includes operations that force a software change of the PC, other than exception-generating operations and direct branch instructions. Some examples of the instructions counted by this event include BR Xn, RET, etc...

Related telemetry artifacts

Metrics

- branch_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x007C ISB_SPEC, Barrier speculatively executed, ISB, event

Counts ISB operations that are executed.

Related telemetry artifacts

Metrics

- barrier_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x007D DSB_SPEC, Barrier speculatively executed, DSB, event

Counts DSB operations that are speculatively issued to Load/Store unit in the CPU.

Related telemetry artifacts

Metrics

- barrier_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x007E DMB_SPEC, Barrier speculatively executed, DMB, event

Counts DMB operations that are speculatively issued to the Load/Store unit in the CPU. This event does not count implied barriers from load acquire/store release operations.

Related telemetry artifacts

Metrics

- barrier_percentage

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x0090 RC_LD_SPEC, Release consistency operation speculatively executed, Load-Acquire, event

Counts any load acquire operations that are speculatively executed. For example: LDAR, LDARH, LDARB

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x0091 RC_ST_SPEC, Release consistency operation speculatively executed, Store-Release, event

Counts any store release operations that are speculatively executed. For example: STLR, STLRH, STLRB

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x8004 SIMD_INST_SPEC, Operation speculatively executed, SIMD, event

Counts speculatively executed operations that are SIMD or SVE vector operations or Advanced SIMD non-scalar operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x8005 ASE_INST_SPEC, Operation speculatively executed, Advanced SIMD, event

Counts speculatively executed Advanced SIMD operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x8040 INT_SPEC, Integer operation speculatively executed, event

Counts speculatively executed integer arithmetic operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

0x8087 PRF_SPEC, Operation speculatively executed, Prefetch, event

Counts speculatively executed operations that prefetch memory. For example: Scalar: PRFM, SVE: PRFB, PRFD, PRFH, or PRFW.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Spec_Operation

6.12 FP_Operation (FP OPERATION) events for Neoverse V3

Speculatively executed floating-point events.

Summary of events in FP_Operation:

- Total implemented Common events: 5
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-12: FP_Operation events summary

Code	Mnemonic	Name	Description
0x8014	FP_HP_SPEC	Floating-point operation speculatively executed, half precision	Counts speculatively executed half precision floating point operations.
0x8018	FP_SP_SPEC	Floating-point operation speculatively executed, single precision	Counts speculatively executed single precision floating point operations.
0x801C	FP_DP_SPEC	Floating-point operation speculatively executed, double precision	Counts speculatively executed double precision floating point operations.
0x80C0	FP_SCALE_OPS_SPEC	Scalable floating-point element ALU operations speculatively executed	Counts speculatively executed scalable single precision floating point operations.
0x80C1	FP_FIXED_OPS_SPEC	Non-scalable floating-point element ALU operations speculatively executed	Counts speculatively executed non-scalable single precision floating point operations.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

`0x8014` FP_HP_SPEC, Floating-point operation speculatively executed, half precision, event

Counts speculatively executed half precision floating point operations.

Related telemetry artifacts

Metrics

- fp16_percentage

Metric groups

FP_Precision_Mix

Functional groups

FP_Operation

`0x8018` FP_SP_SPEC, Floating-point operation speculatively executed, single precision, event

Counts speculatively executed single precision floating point operations.

Related telemetry artifacts

Metrics

- fp32_percentage

Metric groups

FP_Precision_Mix

Functional groups

FP_Operation

`0x801C` FP_DP_SPEC, Floating-point operation speculatively executed, double precision, event

Counts speculatively executed double precision floating point operations.

Related telemetry artifacts

Metrics

- fp64_percentage

Metric groups

FP_Precision_Mix

Functional groups

FP_Operation

`0x80C0` FP_SCALE_OPS_SPEC, Scalable floating-point element ALU operations speculatively executed, event

Counts speculatively executed scalable single precision floating point operations.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Metrics

- sve_fp_ops_per_cycle
- fp_ops_per_cycle

Metric groups

FP_Arithmetic_Intensity

Functional groups

FP_Operation

`0x80C1` FP_FIXED_OPS_SPEC, Non-scalable floating-point element ALU operations speculatively executed, event

Counts speculatively executed non-scalable single precision floating point operations.

Related telemetry artifacts

Metrics

- nonsve_fp_ops_per_cycle
- fp_ops_per_cycle

Metric groups

FP_Arithmetic_Intensity

Functional groups

FP_Operation

6.13 Stall (STALL) events for Neoverse V3

Stall related events.

Summary of events in Stall:

- Total implemented Common events: 21
- Total Implemented Product ImpDef events: 5
- PMU Only events : 5
- ETE Only events : 0

Table 6-13: Stall events summary

Code	Mnemonic	Name	Description
0x0023	STALL_FRONTEND	No operation sent for execution due to the frontend	Counts cycles when frontend could not send any micro-operations to the rename stage because of...
0x0024	STALL_BACKEND	No operation sent for execution due to the backend	Counts cycles whenever the rename unit is unable to send any micro-operations to the backend of...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x003C	STALL	No operation sent for execution	Counts cycles when no operations are sent to the rename unit from the frontend or from the rename...
0x003D	STALL_SLOT_BACKEND	No operation sent for execution on a Slot due to the backend	Counts slots per cycle in which no operations are sent from the rename unit to the backend due to...
0x003E	STALL_SLOT_FRONTEND	No operation sent for execution on a Slot due to the frontend	Counts slots per cycle in which no operations are sent to the rename unit from the frontend due...
0x003F	STALL_SLOT	No operation sent for execution on a Slot	Counts slots per cycle in which no operations are sent to the rename unit from the frontend or...
0x015C	IMP_STALL_BACKEND_IQ_SX	Backend dispatch stall due to no room for operations in simple integer issue queues	Counts the number of dispatch stall cycles due to simple integer issue queues (SX IQ) which...
0x015D	IMP_STALL_BACKEND_IQ_MX	Backend dispatch stall due to no room for operations in complex integer issue queues	Counts the number of dispatch stall cycles due to complex integer issue queues (MX IQ) which...
0x015E	IMP_STALL_BACKEND_IQ_LS	Backend dispatch stall due to no room for operations in load store issue queues	Counts the number of dispatch stall cycles due to load store issue queues (LS IQ) which cannot...
0x015F	IMP_STALL_BACKEND_IQ_VX	Backend dispatch stall due to no room for operations in vector issue queues	Counts the number of dispatch stall cycles due to vector issue queues (VX IQ) which cannot take...
0x0160	IMP_STALL_BACKEND_MCQ	Backend dispatch stage stall, due to full commit queue	Counts cycles where the dispatch stage is stalled because the commit queue (MCQ) is full and...
0x4005	STALL_BACKEND_MEM	Memory stall cycles	Counts cycles when the backend is stalled because there is a demand data miss in the last level...
0x8158	STALL_FRONTEND_MEMBOUND	Frontend stall cycles, memory bound	Counts cycles when the frontend could not send any micro-operations to the rename stage due to...
0x8159	STALL_FRONTEND_L1I	Frontend stall cycles, level 1 instruction cache	Counts cycles when the frontend is stalled because there is an instruction fetch request pending...
0x815B	STALL_FRONTEND_MEM	Frontend stall cycles, last level PE cache or memory	Counts cycles when the frontend is stalled because there is an instruction fetch request pending...
0x815C	STALL_FRONTEND_TLB	Frontend stall cycles, TLB	Counts when the frontend is stalled on any TLB misses being handled. This event also counts the...
0x8160	STALL_FRONTEND_CPUBOUND	Frontend stall cycles, processor bound	Counts cycles when the frontend could not send any micro-operations to the rename stage due to...
0x8161	STALL_FRONTEND_FLOW	Frontend stall cycles, flow control	Counts cycles when the frontend could not send any micro-operations to the rename stage due to...
0x8162	STALL_FRONTEND_FLUSH	Frontend stall cycles, flush recovery	Counts cycles when the frontend could not send any micro-operations to the rename stage as the...
0x8164	STALL_BACKEND_MEMBOUND	Backend stall cycles, memory bound	Counts cycles when the backend could not accept any micro-operations due to resource constraints...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x8165	STALL_BACKEND_L1D	Backend stall cycles, level 1 data cache	Counts cycles when the backend is stalled because there is a demand data miss in the level 1 data...
0x8167	STALL_BACKEND_TLB	Backend stall cycles, TLB	Counts cycles when the backend is stalled on any demand TLB misses being handled.
0x8168	STALL_BACKEND_ST	Backend stall cycles, store	Counts cycles when the backend is stalled and there is a store that has not reached the...
0x816A	STALL_BACKEND_CPUBOUND	Backend stall cycles, processor bound	Counts cycles when the backend could not accept any micro-operations due to any resource...
0x816B	STALL_BACKEND_BUSY	Backend stall cycles, backend busy	Counts cycles when the backend could not accept any micro-operations because the issue queues are...
0x816D	STALL_BACKEND_RENAME	Backend stall cycles, rename full	Counts cycles when backend is stalled even when operations are available from the frontend but at...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0023 STALL_FRONTEND, No operation sent for execution due to the frontend, event

Counts cycles when frontend could not send any micro-operations to the rename stage because of frontend resource stalls caused by fetch memory latency or branch prediction flow stalls. STALL_FRONTEND_SLOTS counts SLOTS during the cycle when this event counts.

Related telemetry artifacts

Metrics

- frontend_stalled_cycles
- frontend_core_bound
- frontend_mem_bound

Metric groups

Cycle_Accounting
Topdown_Frontend

Functional groups

Stall

0x0024 STALL_BACKEND, No operation sent for execution due to the backend, event

Counts cycles whenever the rename unit is unable to send any micro-operations to the backend of the pipeline because of backend resource constraints. Backend resource constraints can include issue stage fullness, execution stage fullness, or other internal pipeline resource fullness. All the backend slots were empty during the cycle when this event counts.

Related telemetry artifacts

Metrics

- backend_stalled_cycles

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- backend_core_bound
- backend_mem_bound
- backend_busy_bound

Metric groups

Cycle_Accounting
Topdown_Backend

Functional groups

Stall

0x003C STALL, No operation sent for execution, event

Counts cycles when no operations are sent to the rename unit from the frontend or from the rename unit to the backend for any reason (either frontend or backend stall). This event is the sum of STALL_FRONTEND and STALL_BACKEND

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Stall

0x003D STALL_SLOT_BACKEND, No operation sent for execution on a Slot due to the backend, event

Counts slots per cycle in which no operations are sent from the rename unit to the backend due to backend resource constraints. STALL_BACKEND counts during the cycle when STALL_SLOT_BACKEND counts at least 1.

Related telemetry artifacts

Metrics

- backend_bound

Metric groups

Topdown_L1

Functional groups

Stall

0x003E STALL_SLOT_FRONTEND, No operation sent for execution on a Slot due to the frontend, event

Counts slots per cycle in which no operations are sent to the rename unit from the frontend due to frontend resource constraints.

Related telemetry artifacts

Metrics

- frontend_bound

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Metric groups

Topdown_L1

Functional groups

Stall

0x003F STALL_SLOT, No operation sent for execution on a Slot, event

Counts slots per cycle in which no operations are sent to the rename unit from the frontend or from the rename unit to the backend for any reason (either frontend or backend stall). STALL_SLOT is the sum of STALL_SLOT_FRONTEND and STALL_SLOT_BACKEND.

Related telemetry artifacts

Metrics

- retiring
- bad_speculation

Metric groups

Topdown_L1

Functional groups

Stall

0x015C IMP_STALL_BACKEND_IQ_SX, Backend dispatch stall due to no room for operations in simple integer issue queues, event

Counts the number of dispatch stall cycles due to simple integer issue queues (SX IQ) which cannot take any operations for execution. This event counts when the oldest operation is waiting to issue to the SX IQ but it is full.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Stall

0x015D IMP_STALL_BACKEND_IQ_MX, Backend dispatch stall due to no room for operations in complex integer issue queues, event

Counts the number of dispatch stall cycles due to complex integer issue queues (MX IQ) which cannot take any operations for execution. This event counts when the oldest operation is waiting to issue to the MX IQ but it is full.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Stall

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x015E IMP_STALL_BACKEND_IQ_LS, Backend dispatch stall due to no room for operations in load store issue queues, event

Counts the number of dispatch stall cycles due to load store issue queues (LS IQ) which cannot take any operations for execution. This event counts when the oldest operation is waiting to issue to the LS IQ but it is full.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Stall

0x015F IMP_STALL_BACKEND_IQ_VX, Backend dispatch stall due to no room for operations in vector issue queues, event

Counts the number of dispatch stall cycles due to vector issue queues (VX IQ) which cannot take any operations for execution. This event counts when the oldest operation is waiting to issue to the VX IQ but it is full.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Stall

0x0160 IMP_STALL_BACKEND_MCQ, Backend dispatch stage stall, due to full commit queue, event

Counts cycles where the dispatch stage is stalled because the commit queue (MCQ) is full and cannot take any operations for execution. The commit queue is responsible for managing the final stage of instruction execution, where instructions are retired in program order after completing execution.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

Stall

0x4005 STALL_BACKEND_MEM, Memory stall cycles, event

Counts cycles when the backend is stalled because there is a demand data miss in the last level core cache.

Related telemetry artifacts

Metrics

- backend_mem_cache_bound
- backend_cache_l1d_bound

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- backend_cache_l2d_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x8158 STALL_FRONTEND_MEMBOUND, Frontend stall cycles, memory bound, event

Counts cycles when the frontend could not send any micro-operations to the rename stage due to resource constraints in the memory resources.

Related telemetry artifacts

Metrics

- frontend_mem_bound
- frontend_mem_cache_bound
- frontend_mem_tlb_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

0x8159 STALL_FRONTEND_L1I, Frontend stall cycles, level 1 instruction cache, event

Counts cycles when the frontend is stalled because there is an instruction fetch request pending in the level 1 instruction cache.

Related telemetry artifacts

Metrics

- frontend_mem_cache_bound
- frontend_cache_l1i_bound
- frontend_cache_l2i_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

0x815B STALL_FRONTEND_MEM, Frontend stall cycles, last level PE cache or memory, event

Counts cycles when the frontend is stalled because there is an instruction fetch request pending in the last level core cache.

Related telemetry artifacts

Metrics

- frontend_mem_cache_bound

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- frontend_cache_l1i_bound
- frontend_cache_l2i_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

0x815C STALL_FRONTEND_TLB, Frontend stall cycles, TLB, event

Counts when the frontend is stalled on any TLB misses being handled. This event also counts the TLB accesses made by hardware prefetches.

Related telemetry artifacts

Metrics

- frontend_mem_tlb_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

0x8160 STALL_FRONTEND_CPUBOUND, Frontend stall cycles, processor bound, event

Counts cycles when the frontend could not send any micro-operations to the rename stage due to resource constraints in the CPU resources excluding memory resources.

Related telemetry artifacts

Metrics

- frontend_core_bound
- frontend_core_flush_bound
- frontend_core_flow_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

0x8161 STALL_FRONTEND_FLOW, Frontend stall cycles, flow control, event

Counts cycles when the frontend could not send any micro-operations to the rename stage due to resource constraints in the branch prediction unit.

Related telemetry artifacts

Metrics

- frontend_core_flow_bound

Metric groups

Topdown_Frontend

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

Stall

0x8162 STALL_FRONTEND_FLUSH, Frontend stall cycles, flush recovery, event

Counts cycles when the frontend could not send any micro-operations to the rename stage as the frontend is recovering from a machine flush or resteer. Example scenarios that cause a flush include branch mispredictions, taken exceptions, micro-architectural flush etc.

Related telemetry artifacts

Metrics

- frontend_bound
- bad_speculation
- frontend_core_flush_bound

Metric groups

Topdown_Frontend

Topdown_L1

Functional groups

Stall

0x8164 STALL_BACKEND_MEMBOUND, Backend stall cycles, memory bound, event

Counts cycles when the backend could not accept any micro-operations due to resource constraints in the memory resources.

Related telemetry artifacts

Metrics

- backend_mem_bound
- backend_mem_cache_bound
- backend_mem_tlb_bound
- backend_mem_store_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x8165 STALL_BACKEND_L1D, Backend stall cycles, level 1 data cache, event

Counts cycles when the backend is stalled because there is a demand data miss in the level 1 data cache.

Related telemetry artifacts

Metrics

- backend_mem_cache_bound
- backend_cache_l1d_bound

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- backend_cache_l2d_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x8167 STALL_BACKEND_TLB, Backend stall cycles, TLB, event

Counts cycles when the backend is stalled on any demand TLB misses being handled.

Related telemetry artifacts

Metrics

- backend_mem_tlb_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x8168 STALL_BACKEND_ST, Backend stall cycles, store, event

Counts cycles when the backend is stalled and there is a store that has not reached the pre-commit stage.

Related telemetry artifacts

Metrics

- backend_mem_store_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x816A STALL_BACKEND_CPUBOUND, Backend stall cycles, processor bound, event

Counts cycles when the backend could not accept any micro-operations due to any resource constraints in the CPU excluding memory resources.

Related telemetry artifacts

Metrics

- backend_core_bound
- backend_core_rename_bound

Metric groups

Topdown_Backend

Functional groups

Stall

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x816B STALL_BACKEND_BUSY, Backend stall cycles, backend busy, event

Counts cycles when the backend could not accept any micro-operations because the issue queues are full to take any operations for execution.

Related telemetry artifacts

Metrics

- backend_busy_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x816D STALL_BACKEND_RENAME, Backend stall cycles, rename full, event

Counts cycles when backend is stalled even when operations are available from the frontend but at least one is not ready to be sent to the backend because no rename register is available.

Related telemetry artifacts

Metrics

- backend_core_rename_bound

Metric groups

Topdown_Backend

Functional groups

Stall

6.14 General (GENERAL) events for Neoverse V3

General CPU related events.

Summary of events in General:

- Total implemented Common events: 2
- Total Implemented Product ImpDef events: 5
- PMU Only events : 5
- ETE Only events : 0

Table 6-14: General events summary

Code	Mnemonic	Name	Description
0x0011	CPU_CYCLES	Cycle	Counts CPU clock cycles (not timer cycles). The clock measured by this event is defined as the...
0x0198	IMP_L2_CHI_RX_CBUSY_0	Received RXDAT or RXRSP responses, with CBusy 0	Counts the number of RXDAT or RXRSP responses received with a CBusy value of 0. Note that if an...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x0199	IMP_L2_CHI_RX_CBUSY_1	Received RXDAT or RXRSP responses, with CBusy 1	Counts the number of RXDAT or RXRSP responses received with a CBusy value of 1. Note that if an...
0x019A	IMP_L2_CHI_RX_CBUSY_2	Received RXDAT or RXRSP responses, with CBusy 2	Counts the number of RXDAT or RXRSP responses received with a CBusy value of 2. Note that if an...
0x019B	IMP_L2_CHI_RX_CBUSY_3	Received RXDAT or RXRSP responses, with CBusy 3	Counts the number of RXDAT or RXRSP responses received with a CBusy value of 3. Note that if an...
0x019C	IMP_L2_CHI_RX_CBUSY_MT	Received RXDAT or RXRSP responses, with CBusy multi-threaded set	Counts the number of RXDAT or RXRSP responses received with CBusy multi-threaded set. Note that...
0x4004	CNT_CYCLES	Constant frequency cycles	Increments at a constant frequency equal to the rate of increment of the System Counter, CNTPCT_EL0.

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0011 CPU_CYCLES, Cycle, event

Counts CPU clock cycles (not timer cycles). The clock measured by this event is defined as the physical clock driving the CPU logic.

Related telemetry artifacts

Metrics

- frontend_stalled_cycles
- backend_stalled_cycles
- frontend_bound
- backend_bound
- retiring
- bad_speculation
- ipc
- sve_fp_ops_per_cycle
- nonsve_fp_ops_per_cycle
- fp_ops_per_cycle

Metric groups

Cycle_Accounting
FP_Arithmetic_Intensity
General
Topdown_L1

Functional groups

General

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x0198 IMP_L2_CHI_RX_CBUSY_0, Received RXDAT or RXRSP responses, with CBusy 0, event

Counts the number of RXDAT or RXRSP responses received with a CBusy value of 0. Note that if an RXDAT flit and RXRSP flit, both with a CBusy value of 0 arrive at the same time this event increments only once.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

General

0x0199 IMP_L2_CHI_RX_CBUSY_1, Received RXDAT or RXRSP responses, with CBusy 1, event

Counts the number of RXDAT or RXRSP responses received with a CBusy value of 1. Note that if an RXDAT flit and RXRSP flit, both with a CBusy value of 1 arrive at the same time this event increments only once.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

General

0x019A IMP_L2_CHI_RX_CBUSY_2, Received RXDAT or RXRSP responses, with CBusy 2, event

Counts the number of RXDAT or RXRSP responses received with a CBusy value of 2. Note that if an RXDAT flit and RXRSP flit, both with a CBusy value of 2 arrive at the same time this event increments only once.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

General

0x019B IMP_L2_CHI_RX_CBUSY_3, Received RXDAT or RXRSP responses, with CBusy 3, event

Counts the number of RXDAT or RXRSP responses received with a CBusy value of 3. Note that if an RXDAT flit and RXRSP flit, both with a CBusy value of 3 arrive at the same time this event increments only once.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

General

0x019C IMP_L2_CHI_RX_CBUSY_MT, Received RXDAT or RXRSP responses, with CBusy multi-threaded set, event

Counts the number of RXDAT or RXRSP responses received with CBusy multi-threaded set. Note that if an RXDAT flit and RXRSP flit, both with a CBusy multi-threaded set arrive at the same time this event increments only once.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

General

0x4004 CNT_CYCLES, Constant frequency cycles, event

Increments at a constant frequency equal to the rate of increment of the System Counter, CNTPCT_EL0.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

General

6.15 TLB (TLB) events for Neoverse V3

TLB and MMU related events.

Summary of events in TLB:

- Total implemented Common events: 34
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-15: TLB events summary

Code	Mnemonic	Name	Description
0x0002	L1I_TLB_REFILL	Level 1 instruction TLB refill	Counts level 1 instruction TLB refills from any Instruction fetch. If there are multiple misses...
0x0005	L1D_TLB_REFILL	Level 1 data TLB refill	Counts level 1 data TLB accesses that resulted in TLB refills. If there are multiple misses in...
0x0025	L1D_TLB	Level 1 data TLB access	Counts level 1 data TLB accesses caused by any memory load or store operation. Note that load or...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x0026	L1I_TLB	Level 1 instruction TLB access	Counts level 1 instruction TLB accesses, whether the access hits or misses in the TLB. This event...
0x002D	L2D_TLB_REFILL	Level 2 data TLB refill	Counts level 2 TLB refills caused by memory operations from both data and instruction fetch,...
0x002F	L2D_TLB	Level 2 data TLB access	Counts level 2 TLB accesses except those caused by TLB maintenance operations.
0x0034	DTLB_WALK	Data TLB access with at least one translation table walk	Counts number of demand data translation table walks caused by a miss in the L2 TLB and...
0x0035	ITLB_WALK	Instruction TLB access with at least one translation table walk	Counts number of instruction translation table walks caused by a miss in the L2 TLB and...
0x004C	L1D_TLB_REFILL_RD	Level 1 data TLB refill, read	Counts level 1 data TLB refills caused by memory read operations. If there are multiple misses in...
0x004D	L1D_TLB_REFILL_WR	Level 1 data TLB refill, write	Counts level 1 data TLB refills caused by data side memory write operations. If there are...
0x004E	L1D_TLB_RD	Level 1 data TLB access, read	Counts level 1 data TLB accesses caused by memory read operations. This event counts whether the...
0x004F	L1D_TLB_WR	Level 1 data TLB access, write	Counts any L1 data side TLB accesses caused by memory write operations. This event counts whether...
0x005C	L2D_TLB_REFILL_RD	Level 2 data TLB refill, read	Counts level 2 TLB refills caused by memory read operations from both data and instruction fetch...
0x005D	L2D_TLB_REFILL_WR	Level 2 data TLB refill, write	Counts level 2 TLB refills caused by memory write operations from both data and instruction fetch...
0x005E	L2D_TLB_RD	Level 2 data TLB access, read	Counts level 2 TLB accesses caused by memory read operations from both data and instruction fetch...
0x005F	L2D_TLB_WR	Level 2 data TLB access, write	Counts level 2 TLB accesses caused by memory write operations from both data and instruction...
0x8128	DTLB_WALK_PERCYC	Event in progress, DTLB WALK	Counts the number of data translation table walks in progress per cycle.
0x8129	ITLB_WALK_PERCYC	Event in progress, ITLB WALK	Counts the number of instruction translation table walks in progress per cycle.
0x8130	L1D_TLB_RW	Level 1 data TLB demand access	Counts level 1 data TLB demand accesses caused by memory read or write operations. This event...
0x8131	L1I_TLB_RD	Level 1 instruction TLB demand access	Counts level 1 instruction TLB demand accesses whether the access hits or misses in the TLB.
0x8132	L1D_TLB_PRFM	Level 1 data TLB software preload	Counts level 1 data TLB accesses generated by software prefetch or preload memory accesses. Load...
0x8133	L1I_TLB_PRFM	Level 1 instruction TLB software preload	Counts level 1 instruction TLB accesses generated by software preload or prefetch instructions....
0x8134	DTLB_HWUPD	Data TLB hardware update of translation table	Counts number of memory accesses triggered by a data translation table walk and performing an...
0x8135	ITLB_HWUPD	Instruction TLB hardware update of translation table	Counts number of memory accesses triggered by an instruction translation table walk and...
0x8136	DTLB_STEP	Data TLB translation table walk, step	Counts number of memory accesses triggered by a demand data translation table walk and performing...
0x8137	ITLB_STEP	Instruction TLB translation table walk, step	Counts number of memory accesses triggered by an instruction translation table walk and...

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x8138	DTLB_WALK_LARGE	Data TLB large page translation table walk	Counts number of demand data translation table walks caused by a miss in the L2 TLB and yielding...
0x8139	ITLB_WALK_LARGE	Instruction TLB large page translation table walk	Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding...
0x813A	DTLB_WALK_SMALL	Data TLB small page translation table walk	Counts number of data translation table walks caused by a miss in the L2 TLB and yielding a small...
0x813B	ITLB_WALK_SMALL	Instruction TLB small page translation table walk	Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding...
0x813C	DTLB_WALK_RW	Data TLB demand access with at least one translation table walk	Counts number of demand data translation table walks caused by a miss in the L2 TLB and...
0x813D	ITLB_WALK_RD	Instruction TLB demand access with at least one translation table walk	Counts number of demand instruction translation table walks caused by a miss in the L2 TLB and...
0x813E	DTLB_WALK_PRFM	Data TLB software preload access with at least one translation table walk	Counts number of software prefetches or preloads generated data translation table walks caused by...
0x813F	ITLB_WALK_PRFM	Instruction TLB software preload access with at least one translation table walk	Counts number of software prefetches or preloads generated instruction translation table walks...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0002 L1I_TLB_REFILL, Level 1 instruction TLB refill, event

Counts level 1 instruction TLB refills from any Instruction fetch. If there are multiple misses in the TLB that are resolved by the refill, then this event only counts once. This event will not count if the translation table walk results in a fault (such as a translation or access fault), since there is no new translation created for the TLB.

Related telemetry artifacts

Metrics

- l1i_tlb_mpki in ITLB_Effectiveness
- l1i_tlb_mpki in MPKI
- l1i_tlb_miss_ratio in ITLB_Effectiveness
- l1i_tlb_miss_ratio in Miss_Ratio

Metric groups

ITLB_Effectiveness
MPKI
Miss_Ratio

Functional groups

TLB

0x0005 L1D_TLB_REFILL, Level 1 data TLB refill, event

Counts level 1 data TLB accesses that resulted in TLB refills. If there are multiple misses in the TLB that are resolved by the refill, then this event only counts once. This event counts for refills caused by preload instructions or hardware prefetch accesses. This event counts regardless of whether the miss hits in L2 or results in a translation table walk. This event will not count if the

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

translation table walk results in a fault (such as a translation or access fault), since there is no new translation created for the TLB. This event will not count on an access from an AT(address translation) instruction.

Related telemetry artifacts

Metrics

- l1d_tlb_mpki in DTLB_Effectiveness
- l1d_tlb_mpki in MPKI
- l1d_tlb_miss_ratio in DTLB_Effectiveness
- l1d_tlb_miss_ratio in Miss_Ratio

Metric groups

DTLB_Effectiveness
MPKI
Miss_Ratio

Functional groups

TLB

`0x0025` L1D_TLB, Level 1 data TLB access, event

Counts level 1 data TLB accesses caused by any memory load or store operation. Note that load or store instructions can be broken up into multiple memory operations. This event does not count TLB maintenance operations.

Related telemetry artifacts

Metrics

- dtlb_walk_ratio in DTLB_Effectiveness
- dtlb_walk_ratio in Miss_Ratio
- l1d_tlb_miss_ratio in DTLB_Effectiveness
- l1d_tlb_miss_ratio in Miss_Ratio

Metric groups

DTLB_Effectiveness
Miss_Ratio

Functional groups

TLB

`0x0026` L1I_TLB, Level 1 instruction TLB access, event

Counts level 1 instruction TLB accesses, whether the access hits or misses in the TLB. This event counts both demand accesses and prefetch or preload generated accesses.

Related telemetry artifacts

Metrics

- itlb_walk_ratio in ITLB_Effectiveness

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- itlb_walk_ratio in Miss_Ratio
- l1i_tlb_miss_ratio in ITLB_Effectiveness
- l1i_tlb_miss_ratio in Miss_Ratio

Metric groups

ITLB_Effectiveness
Miss_Ratio

Functional groups

TLB

0x002D L2D_TLB_REFILL, Level 2 data TLB refill, event

Counts level 2 TLB refills caused by memory operations from both data and instruction fetch, except for those caused by TLB maintenance operations and hardware prefetches.

Related telemetry artifacts

Metrics

- l2_tlb_mpki in DTLB_Effectiveness
- l2_tlb_mpki in ITLB_Effectiveness
- l2_tlb_mpki in MPKI
- l2_tlb_miss_ratio in DTLB_Effectiveness
- l2_tlb_miss_ratio in ITLB_Effectiveness
- l2_tlb_miss_ratio in Miss_Ratio

Metric groups

DTLB_Effectiveness
ITLB_Effectiveness
MPKI
Miss_Ratio

Functional groups

TLB

0x002F L2D_TLB, Level 2 data TLB access, event

Counts level 2 TLB accesses except those caused by TLB maintenance operations.

Related telemetry artifacts

Metrics

- l2_tlb_miss_ratio in DTLB_Effectiveness
- l2_tlb_miss_ratio in ITLB_Effectiveness
- l2_tlb_miss_ratio in Miss_Ratio

Metric groups

DTLB_Effectiveness
ITLB_Effectiveness

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Miss_Ratio

Functional groups

TLB

0x0034 DTLB_WALK, Data TLB access with at least one translation table walk, event

Counts number of demand data translation table walks caused by a miss in the L2 TLB and performing at least one memory access. Translation table walks are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD. Note that partial translations that cause a translation table walk are also counted. Also note that this event counts walks triggered by software preloads, but not walks triggered by hardware prefetchers, and that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

Metrics

- dtlb_mpki in DTLB_Effectiveness
- dtlb_mpki in MPKI
- dtlb_walk_ratio in DTLB_Effectiveness
- dtlb_walk_ratio in Miss_Ratio

Metric groups

DTLB_Effectiveness
MPKI
Miss_Ratio

Functional groups

TLB

0x0035 ITLB_WALK, Instruction TLB access with at least one translation table walk, event

Counts number of instruction translation table walks caused by a miss in the L2 TLB and performing at least one memory access. Translation table walks are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

Metrics

- itlb_mpki in ITLB_Effectiveness
- itlb_mpki in MPKI
- itlb_walk_ratio in ITLB_Effectiveness
- itlb_walk_ratio in Miss_Ratio

Metric groups

ITLB_Effectiveness
MPKI

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Miss_Ratio

Functional groups

TLB

0x004C L1D_TLB_REFILL_RD, Level 1 data TLB refill, read, event

Counts level 1 data TLB refills caused by memory read operations. If there are multiple misses in the TLB that are resolved by the refill, then this event only counts once. This event counts for refills caused by preload instructions or hardware prefetch accesses. This event counts regardless of whether the miss hits in L2 or results in a translation table walk. This event will not count if the translation table walk results in a fault (such as a translation or access fault), since there is no new translation created for the TLB. This event will not count on an access from an Address Translation (AT) instruction.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x004D L1D_TLB_REFILL_WR, Level 1 data TLB refill, write, event

Counts level 1 data TLB refills caused by data side memory write operations. If there are multiple misses in the TLB that are resolved by the refill, then this event only counts once. This event counts for refills caused by preload instructions or hardware prefetch accesses. This event counts regardless of whether the miss hits in L2 or results in a translation table walk. This event will not count if the table walk results in a fault (such as a translation or access fault), since there is no new translation created for the TLB. This event will not count with an access from an Address Translation (AT) instruction.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x004E L1D_TLB_RD, Level 1 data TLB access, read, event

Counts level 1 data TLB accesses caused by memory read operations. This event counts whether the access hits or misses in the TLB. This event does not count TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x004F L1D_TLB_WR, Level 1 data TLB access, write, event

Counts any L1 data side TLB accesses caused by memory write operations. This event counts whether the access hits or misses in the TLB. This event does not count TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x005C L2D_TLB_REFILL_RD, Level 2 data TLB refill, read, event

Counts level 2 TLB refills caused by memory read operations from both data and instruction fetch except for those caused by TLB maintenance operations or hardware prefetches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x005D L2D_TLB_REFILL_WR, Level 2 data TLB refill, write, event

Counts level 2 TLB refills caused by memory write operations from both data and instruction fetch except for those caused by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x005E L2D_TLB_RD, Level 2 data TLB access, read, event

Counts level 2 TLB accesses caused by memory read operations from both data and instruction fetch except for those caused by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x005F L2D_TLB_WR, Level 2 data TLB access, write, event

Counts level 2 TLB accesses caused by memory write operations from both data and instruction fetch except for those caused by TLB maintenance operations.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

`0x8128` DTLB_WALK_PERCYC, Event in progress, DTLB WALK, event

Counts the number of data translation table walks in progress per cycle.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

`0x8129` ITLB_WALK_PERCYC, Event in progress, ITLB WALK, event

Counts the number of instruction translation table walks in progress per cycle.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

`0x8130` L1D_TLB_RW, Level 1 data TLB demand access, event

Counts level 1 data TLB demand accesses caused by memory read or write operations. This event counts whether the access hits or misses in the TLB. This event does not count TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

`0x8131` L1I_TLB_RD, Level 1 instruction TLB demand access, event

Counts level 1 instruction TLB demand accesses whether the access hits or misses in the TLB.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x8132 L1D_TLB_PRFM, Level 1 data TLB software preload, event

Counts level 1 data TLB accesses generated by software prefetch or preload memory accesses. Load or store instructions can be broken into multiple memory operations. This event does not count TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x8133 L1I_TLB_PRFM, Level 1 instruction TLB software preload, event

Counts level 1 instruction TLB accesses generated by software preload or prefetch instructions. This event counts whether the access hits or misses in the TLB. This event does not count TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x8134 DTLB_HWUPD, Data TLB hardware update of translation table, event

Counts number of memory accesses triggered by a data translation table walk and performing an update of a translation table entry. Memory accesses are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD. Note that this event counts accesses triggered by software preloads, but not accesses triggered by hardware prefetchers.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x8135 ITLB_HWUPD, Instruction TLB hardware update of translation table, event

Counts number of memory accesses triggered by an instruction translation table walk and performing an update of a translation table entry. Memory accesses are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x8136 DTLB_STEP, Data TLB translation table walk, step, event

Counts number of memory accesses triggered by a demand data translation table walk and performing a read of a translation table entry. Memory accesses are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD. Note that this event counts accesses triggered by software preloads, but not accesses triggered by hardware prefetchers.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x8137 ITLB_STEP, Instruction TLB translation table walk, step, event

Counts number of memory accesses triggered by an instruction translation table walk and performing a read of a translation table entry. Memory accesses are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x8138 DTLB_WALK_LARGE, Data TLB large page translation table walk, event

Counts number of demand data translation table walks caused by a miss in the L2 TLB and yielding a large page. The set of large pages is defined as all pages with a final size higher than or equal to 2MB. Translation table walks that end up taking a translation fault are not counted, as the page size would be undefined in that case. If DTLB_WALK_BLOCK is implemented, then it is an alias for this event in this family. Note that partial translations that cause a translation table walk are also counted. Also note that this event counts walks triggered by software preloads, but not walks triggered by hardware prefetchers, and that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x8139 ITLB_WALK_LARGE, Instruction TLB large page translation table walk, event

Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding a large page. The set of large pages is defined as all pages with a final size higher than or equal to 2MB. Translation table walks that end up taking a translation fault are not counted, as the page size would be undefined in that case. In this family, this is equal to ITLB_WALK_BLOCK event. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x813A DTLB_WALK_SMALL, Data TLB small page translation table walk, event

Counts number of data translation table walks caused by a miss in the L2 TLB and yielding a small page. The set of small pages is defined as all pages with a final size lower than 2MB. Translation table walks that end up taking a translation fault are not counted, as the page size would be undefined in that case. If DTLB_WALK_PAGE event is implemented, then it is an alias for this event in this family. Note that partial translations that cause a translation table walk are also counted. Also note that this event counts walks triggered by software preloads, but not walks triggered by hardware prefetchers, and that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x813B ITLB_WALK_SMALL, Instruction TLB small page translation table walk, event

Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding a small page. The set of small pages is defined as all pages with a final size lower than 2MB. Translation table walks that end up taking a translation fault are not counted, as the page size would be undefined in that case. In this family, this is equal to ITLB_WALK_PAGE event. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x813C DTLB_WALK_RW, Data TLB demand access with at least one translation table walk, event

Counts number of demand data translation table walks caused by a miss in the L2 TLB and performing at least one memory access. Translation table walks are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional groups

TLB

0x813D ITLB_WALK_RD, Instruction TLB demand access with at least one translation table walk, event

Counts number of demand instruction translation table walks caused by a miss in the L2 TLB and performing at least one memory access. Translation table walks are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x813E DTLB_WALK_PRFM, Data TLB software preload access with at least one translation table walk, event

Counts number of software prefetches or preloads generated data translation table walks caused by a miss in the L2 TLB and performing at least one memory access. Translation table walks are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

0x813F ITLB_WALK_PRFM, Instruction TLB software preload access with at least one translation table walk, event

Counts number of software prefetches or preloads generated instruction translation table walks caused by a miss in the L2 TLB and performing at least one memory access. Translation table walks are counted even if the translation ended up taking a translation fault for reasons different than EPD, E0PD and NFD. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

TLB

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.16 SVE (SVE) events for Neoverse V3

SVE related events.

Summary of events in SVE:

- Total implemented Common events: 12
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-16: SVE events summary

Code	Mnemonic	Name	Description
0x8006	SVE_INST_SPEC	Operation speculatively executed, SVE, including load and store	Counts speculatively executed operations that are SVE operations.
0x8074	SVE_PRED_SPEC	Operation speculatively executed, SVE predicated	Counts speculatively executed predicated SVE operations.
0x8075	SVE_PRED_EMPTY_SPEC	Operation speculatively executed, SVE predicated with no active predicates	Counts speculatively executed predicated SVE operations with no active predicate elements.
0x8076	SVE_PRED_FULL_SPEC	Operation speculatively executed, SVE predicated with all active predicates	Counts speculatively executed predicated SVE operations with all predicate elements active.
0x8077	SVE_PRED_PARTIAL_SPEC	Operation speculatively executed, SVE predicated with partially active predicates	Counts speculatively executed predicated SVE operations with at least one but not all active...
0x8079	SVE_PRED_NOT_FULL_SPEC	SVE predicated operations speculatively executed with no active or partially active predicates	Counts speculatively executed predicated SVE operations with at least one non active predicate...
0x80BC	SVE_LDFF_SPEC	Operation speculatively executed, SVE first-fault load	Counts speculatively executed SVE first fault or non-fault load operations.
0x80BD	SVE_LDFF_FAULT_SPEC	Operation speculatively executed, SVE first-fault load which set FFR bit to 0b0	Counts speculatively executed SVE first fault or non-fault load operations that clear at least...
0x80E3	ASE_SVE_INT8_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 8-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80E7	ASE_SVE_INT16_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 16-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80EB	ASE_SVE_INT32_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 32-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80EF	ASE_SVE_INT64_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 64-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x8006 SVE_INST_SPEC, Operation speculatively executed, SVE, including load and store, event

Counts speculatively executed operations that are SVE operations.

Related telemetry artifacts

Metrics

- sve_all_percentage

Metric groups

Operation_Mix

Functional groups

SVE

0x8074 SVE_PRED_SPEC, Operation speculatively executed, SVE predicated, event

Counts speculatively executed predicated SVE operations.

Related telemetry artifacts

Metrics

- sve_predicate_percentage
- sve_predicate_full_percentage
- sve_predicate_partial_percentage
- sve_predicate_empty_percentage

Metric groups

SVE_Effectiveness

Functional groups

SVE

0x8075 SVE_PRED_EMPTY_SPEC, Operation speculatively executed, SVE predicated with no active predicates, event

Counts speculatively executed predicated SVE operations with no active predicate elements.

Related telemetry artifacts

Metrics

- sve_predicate_empty_percentage

Metric groups

SVE_Effectiveness

Functional groups

SVE

0x8076 SVE_PRED_FULL_SPEC, Operation speculatively executed, SVE predicated with all active predicates, event

Counts speculatively executed predicated SVE operations with all predicate elements active.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

Metrics

- sve_predicate_full_percentage

Metric groups

SVE_Effectiveness

Functional groups

SVE

0x8077 SVE_PRED_PARTIAL_SPEC, Operation speculatively executed, SVE predicated with partially active predicates, event

Counts speculatively executed predicated SVE operations with at least one but not all active predicate elements.

Related telemetry artifacts

Metrics

- sve_predicate_partial_percentage

Metric groups

SVE_Effectiveness

Functional groups

SVE

0x8079 SVE_PRED_NOT_FULL_SPEC, SVE predicated operations speculatively executed with no active or partially active predicates, event

Counts speculatively executed predicated SVE operations with at least one non active predicate elements.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SVE

0x80BC SVE_LDFF_SPEC, Operation speculatively executed, SVE first-fault load, event

Counts speculatively executed SVE first fault or non-fault load operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SVE

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x80BD SVE_LDFF_FAULT_SPEC, Operation speculatively executed, SVE first-fault load which set FFR bit to 0b0, event

Counts speculatively executed SVE first fault or non-fault load operations that clear at least one bit in the FFR.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SVE

0x80E3 ASE_SVE_INT8_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 8-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type an 8-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SVE

0x80E7 ASE_SVE_INT16_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 16-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 16-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SVE

0x80EB ASE_SVE_INT32_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 32-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 32-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SVE

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x80EF ASE_SVE_INT64_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 64-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 64-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

SVE

6.17 BRBE (BRBE) events for Neoverse V3

BRBE related events.

Summary of events in BRBE:

- Total implemented Common events: 1
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-17: BRBE events summary

Code	Mnemonic	Name	Description
0x811F	BRB_FILTRATE	Branch Record captured	Counts branch records captured which are not removed by filtering.

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x811F BRB_FILTRATE, Branch Record captured, event

Counts branch records captured which are not removed by filtering.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

BRBE

6.18 CPU_Debug (CPU_DEBUG) events for Neoverse V3

CPU performance debug related events.

Summary of events in CPU_Debug:

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Total implemented Common events: 0
- Total Implemented Product ImpDef events: 15
- PMU Only events : 15
- ETE Only events : 0

Table 6-18: CPU_Debug events summary

Code	Mnemonic	Name	Description
0x0108	IMP_L2_CACHE_REFILL_IF	Allocated cache lines to L2 cache due to instruction fetches	Counts the number of cache lines that are allocated to the L2 cache due to an instruction...
0x010B	IMP_L2_CACHE_PREFETCH_LATE	Late prefetch requests to L2 cache	Counts level 2 cache demand accesses for which prefetch requests were late. This event counts any...
0x0120	IMP_CT_FLUSH	Pipeline flushes in the commit unit	Counts the number of pipeline flushes in the commit unit. These flushes refer to architectural...
0x0121	IMP_CT_FLUSH_MEM_HAZARD	Non-speculative pipeline flushes in the commit unit due to memory hazards	Counts the number of non-speculative pipeline flushes in the commit unit caused by memory...
0x0122	IMP_CT_FLUSH_BAD_BRANCH	Pipeline flushes in the commit unit due to non-branch instructions predicted as branch	Counts the number of pipeline flushes in the commit unit caused by a non-branch instruction...
0x0123	IMP_CT_FLUSH_PREDECODE_ERR	Pipeline flushes in the commit unit due to bad precode	Counts the number of pipeline flushes in the commit unit due to bad predecode. Predecoding is an...
0x0124	IMP_CT_FLUSH_ISB	Pipeline flushes in the commit unit due to Instruction Synchronization Barrier or similar	Counts the number of pipeline flushes in the commit unit caused by an Instruction Synchronization...
0x0125	IMP_CT_FLUSH_OTHER	Pipeline flushes in the commit unit due to other hazards	Counts the number of pipeline flushes in the commit unit that are caused by rare instruction...
0x0127	IMP_LS_RAR_HAZARD	Generated load store hazards due to a Read-After-Read ordering hazard	Counts any load store detected hazards that are generated due to a Read-After -Read (RAR)...
0x0128	IMP_LS_RAW_HAZARD	Generated load store hazards due to a Read-After-Write ordering hazard	Counts any load store detected hazards that are generated due to a Read-After-Write (RAW)...
0x0158	IMP_STALL_BACKEND_RENAME_FRF	Backend rename stall due to no available flag registers	Counts the number of backend rename stall cycles due to the flag registers being full (FRF), that...
0x0159	IMP_STALL_BACKEND_RENAME_GRF	Backend rename stall due to no available general purpose registers	Counts the number of backend rename stall cycles due to the general purpose registers being full...
0x015A	IMP_STALL_BACKEND_RENAME_VRF	Backend rename stall due to no available vector registers	Counts the number of backend rename stall cycles due to the vector registers being full (VRF),...
0x017B	IMP_NEAR_CAS	CAS instruction locally executed	Counts CAS instructions that executed locally to the PE.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Code	Mnemonic	Name	Description
0x017C	IMP_NEAR_CAS_PASS	CAS instruction locally executed with local cache updates	Counts CAS instructions that executed locally to the PE and updated in the local cache.

For a complete list of the events in Neoverse V3, see PMU events cheat sheet for Neoverse V3 and PMU events lookup table for Neoverse V3.

0x0108 IMP_L2_CACHE_REFILL_IF, Allocated cache lines to L2 cache due to instruction fetches, event

Counts the number of cache lines that are allocated to the L2 cache due to an instruction fetch. This event may help indicate that the following mechanisms can improve execution fetch and efficiency: Code layout optimizations, Instruction prefetching, Increased L1I cache size

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

0x010B IMP_L2_CACHE_PREFETCH_LATE, Late prefetch requests to L2 cache, event

Counts level 2 cache demand accesses for which prefetch requests were late. This event counts any lookups in the L2 cache that occur after a prefetch request was generated by the prefetcher, but before the prefetched cache line is allocated into the cache. It indicates that the prefetch was useful but not on time.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

0x0120 IMP_CT_FLUSH, Pipeline flushes in the commit unit, event

Counts the number of pipeline flushes in the commit unit. These flushes refer to architectural and microarchitectural flushes that are resolved when an instruction is no longer speculative. It also includes speculative branch redirects for mispredicted branches. These redirects are not necessarily full pipeline flushes, and they may occur before the branch is the oldest instruction in flight.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

0x0121 IMP_CT_FLUSH_MEM_HAZARD, Non-speculative pipeline flushes in the commit unit due to memory hazards, event

Counts the number of non-speculative pipeline flushes in the commit unit caused by memory hazards, including Read-After-Write, Write-After- Read, and Write-After-Write hazards.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

`0x0122` IMP_CT_FLUSH_BAD_BRANCH, Pipeline flushes in the commit unit due to non-branch instructions predicted as branch, event

Counts the number of pipeline flushes in the commit unit caused by a non-branch instruction predicted as a branch. This type of flush occurs when the branch predictor incorrectly identifies a regular instruction as a branch, which causes speculative execution to follow an incorrect control flow path. When the CPU later detects this misprediction, it must flush the pipeline and restart execution from the correct instruction.Counts flushes due to non-branch instructions predicted as a branch.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

`0x0123` IMP_CT_FLUSH_PREDECODE_ERR, Pipeline flushes in the commit unit due to bad precode, event

Counts the number of pipeline flushes in the commit unit due to bad predecode. Predecoding is an early-stage pipeline process where fetched instructions are analyzed and prepared for execution. Errors in this stage can occur due to corrupted instruction fetches, incorrect decoding of complex instructions, or invalid instruction formats. When such errors are detected, the pipeline must be flushed and restarted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

`0x0124` IMP_CT_FLUSH_ISB, Pipeline flushes in the commit unit due to Instruction Synchronization Barrier or similar, event

Counts the number of pipeline flushes in the commit unit caused by an Instruction Synchronization Barrier (ISB) or an instruction that causes similar side effects. An ISB is a barrier instruction used to ensure that all previous instructions have completed before any subsequent instructions are fetched and executed. It is commonly used when switching execution contexts, modifying system registers, or updating the instruction-related state, for example, after self-modifying code. ISBs are necessary for correctness. However, they introduce pipeline stalls, because they force instruction fetch and decode stages to restart.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

`0x0125` IMP_CT_FLUSH_OTHER, Pipeline flushes in the commit unit due to other hazards, event

Counts the number of pipeline flushes in the commit unit that are caused by rare instruction sequences or microarchitecture states.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

`0x0127` IMP_LS_RAR_HAZARD, Generated load store hazards due to a Read-After-Read ordering hazard, event

Counts any load store detected hazards that are generated due to a Read-After -Read (RAR) ordering hazard. This hazard occurs when a load operation is incorrectly speculated or executed out-of-order relative to an earlier load. It leads to potential data inconsistencies. To maintain memory ordering correctness, the CPU invalidates the speculatively executed load instructions and reissues them in the correct order. It flushes the pipeline and execution stalls.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

`0x0128` IMP_LS_RAW_HAZARD, Generated load store hazards due to a Read-After-Write ordering hazard, event

Counts any load store detected hazards that are generated due to a Read-After-Write (RAW) ordering hazard. This hazard occurs when a younger load instruction executes before an older store to the same memory location. It leads to incorrect data being read. The CPU detects such violations and triggers a load store flush. It clears the pipeline and re-executes affected instructions to ensure a correct execution order.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

0x0158 IMP_STALL_BACKEND_RENAME_FRF, Backend rename stall due to no available flag registers, event

Counts the number of backend rename stall cycles due to the flag registers being full (FRF), that is, no flag registers are available. This event counts when an operation is available to be sent to the backend but cannot be sent because all physical flag registers are in use. Flag registers store condition codes such as zero, carry, overflow, and negative flags. These registers are used for conditional execution, comparisons, and arithmetic operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

0x0159 IMP_STALL_BACKEND_RENAME_GRF, Backend rename stall due to no available general purpose registers, event

Counts the number of backend rename stall cycles due to the general purpose registers being full (GRF), that is, no general purpose registers are available. This event counts when an operation is available to be sent to the backend but cannot be sent because all physical general purpose registers are in use.i These registers store temporary data operands and computational results. Their availability is crucial to sustain instruction throughput.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

0x015A IMP_STALL_BACKEND_RENAME_VRF, Backend rename stall due to no available vector registers, event

Counts the number of backend rename stall cycles due to the vector registers being full (VRF), that is, no vector registers are available. This event counts when an operation is available to be sent to the backend but cannot be sent because all physical vector registers are in use. Vector registers store Single Instruction, Multiple DATA (SIMD) and Scalable Vector Extension (SVE) operations. Their availability is critical to accelerate parallel computations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

0x017B IMP_NEAR_CAS, CAS instruction locally executed, event

Counts CAS instructions that executed locally to the PE.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

0x017C IMP_NEAR_CAS_PASS, CAS instruction locally executed with local cache updates, event

Counts CAS instructions that executed locally to the PE and updated in the local cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse V3 Core Specification.

Functional groups

CPU_Debug

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

7. Supplemental performance debug PMU events

Neoverse V3 implements an additional set of events that are available for use in debugging the performance behaviors of the processor. These events are not guaranteed to have the same level of accuracy as the architected PMU counters. Any use of these events must take this variable accuracy into account.

The following table shows the events.

Table 7-1: Supplemental performance debug PMU events for Neoverse V3

Code	Mnemonic	Description
0x108	IMP_L2_CACHE_IF_REFILL	Level 2 cache refill, fetch
0x10B	IMP_L2_CACHE_PF_LATE_REFILL	Level 2 prefetch requests, late
0x120	IMP_CT_FLUSH	Flushes including architectural, microarchitectural, and branch redirects
0x121	IMP_CT_FLUSH_MEM	Flushes due to memory hazards
0x122	IMP_CT_FLUSH_BAD_BRANCH	Flushes due to non-branch instruction predicted as a branch
0x123	IMP_CT_FLUSH_PREDECODE_ERR	Flushes due to bad predecode
0x124	IMP_CT_FLUSH_ISB	Flushes due to ISB or similar side-effects
0x125	IMP_CT_FLUSH_OTHER	Flushes due to other hazards
0x127	IMP_LS_RAR	Loadstore detected nuke due to read-after-read ordering hazard
0x128	IMP_LS_RAW	Loadstore detected nuke due to read-after-write ordering hazard
0x158	IMP_STALL_BACKEND_RENAME_FRF	RN dispatch stall due to flag registers
0x159	IMP_STALL_BACKEND_RENAME_GRF	RN dispatch stall due to general registers
0x15A	IMP_STALL_BACKEND_RENAME_VRF	RN dispatch stall due to vector registers
0x17B	IMP_NEAR_CAS	Near atomics: compare and swap
0x17C	IMP_NEAR_CAS_PAS	Near atomics: compare and swap pass

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm’s trademark usage guidelines at https://www.arm.com/company/policies/trademarks. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

The information in this document is Final, that is for a developed product.

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
0200-02	15 August 2025	Non-Confidential	Second issue for all revisions of the product.
0100-01	23 September 2024	Non-Confidential	First issue for all revisions of the product.

Change history

The Change history tables describe the technical changes between released issues of this document in reverse order. Issue numbers match the revision history in Document release information on page 191.

Table 2: Differences between issues 0200-02 and 0100-01

Change	Location
Added CPU_CYCLES to the Floating Point Arithmetic Intensity group	PMU events cheat sheet for Neoverse V3
Added metrics to CPU_CYCLES: sve_fp_ops_per_cycle, nonsve_fp_ops_per_cycle, and fp_ops_per_cycle	PMU events lookup table for Neoverse V3

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Change	Location
Added functional group CPU_Debug	PMU events lookup table for Neoverse V3
Edited some events naming	PMU events lookup table for Neoverse V3
Removed 0x8166 STALL_BACKEND_L2D and 0x816C STALL_BACKEND_ILOCK	PMU events lookup table for Neoverse V3
Added 0x81BD L2D_CACHE_REFILL_HWPRF	PMU events lookup table for Neoverse V3
Updated backend_mem_store_bound	Topdown_Backend metrics for Neoverse V3
Updated sve_fp_ops_per_cycle, nonsve_fp_ops_per_cycle, and fp_ops_per_cycle	FP_Arithmetic_Intensity metrics for Neoverse V3
Updated the number of events	PMU events by functional group in Neoverse V3
Added the CPU_Debug group	PMU events by functional group in Neoverse V3
Updated the number of Total implemented Common events	L2_Cache (L2 CACHE) events for Neoverse V3
Updated events 0x01B8 IMP_L2D_CACHE_L1HWPRF and 0x01B9 IMP_L2D_CACHE_REFILL_L1HWPRF	L2_Cache (L2 CACHE) events for Neoverse V3
Added 0x81BD L2D_CACHE_REFILL_HWPRF	L2_Cache (L2 CACHE) events for Neoverse V3
Updated descriptions for 0x0036 LL_CACHE_RD and 0x0037 LL_CACHE_MISS_RD	LL_Cache (LL CACHE) events for Neoverse V3
Updated the number of Total implemented Common events	Stall (STALL) events for Neoverse V3
Renamed 0x015C DISPATCH_STALL_IQ_SX to IMP_STALL_BACKEND_IQ_SX and reworded the name and description	Stall (STALL) events for Neoverse V3
Renamed 0x015D DISPATCH_STALL_IQ_MX to IMP_STALL_BACKEND_IQ_MX and reworded the name and description	Stall (STALL) events for Neoverse V3
Renamed 0x015E event DISPATCH_STALL_IQ_LS to IMP_STALL_BACKEND_IQ_LS and reworded the name and description	Stall (STALL) events for Neoverse V3
Renamed 0x015F event DISPATCH_STALL_IQ_VX to IMP_STALL_BACKEND_IQ_VX and reworded the name and description	Stall (STALL) events for Neoverse V3
Renamed 0x0160 event DISPATCH_STALL_MCQ to IMP_STALL_BACKEND_MCQ and reworded the name and description	Stall (STALL) events for Neoverse V3
Updated description for 0x4005 STALL_BACKEND_MEM	Stall (STALL) events for Neoverse V3
Removed 0x8166 STALL_BACKEND_L2D	Stall (STALL) events for Neoverse V3
Removed 0x816C STALL_BACKEND_ILOCK	Stall (STALL) events for Neoverse V3
Renamed 0x0198 L2_CHI_CBUSY0 to IMP_L2_CHI_RX_CBUSY_0 and reworded the name and description	General (GENERAL) events for Neoverse V3
Renamed 0x0199 L2_CHI_CBUSY1 to IMP_L2_CHI_RX_CBUSY_1 and reworded the name and description	General (GENERAL) events for Neoverse V3

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Change	Location
Renamed 0x019A L2_CHI_CBUSY2 to IMP_L2_CHI_RX_CBUSY_2 and reworded the name and description	General (GENERAL) events for Neoverse V3
Renamed 0x019B L2_CHI_CBUSY3 to IMP_L2_CHI_RX_CBUSY_3 and reworded the name and description	General (GENERAL) events for Neoverse V3
Renamed 0x019C L2_CHI_CBUSY_MT to IMP_L2_CHI_RX_CBUSY_MT and reworded the name and description	General (GENERAL) events for Neoverse V3
Changed Events without a functional group to CPU_Debug (CPU_DEBUG) events	CPU_Debug (CPU_DEBUG) events for Neoverse V3
Reworded all names and descriptions	CPU_Debug (CPU_DEBUG) events for Neoverse V3
Renamed 0x0108 IMP_L2_CACHE_IF_REFILL to IMP_L2_CACHE_REFILL_IF	CPU_Debug (CPU_DEBUG) events for Neoverse V3
Renamed 0x0121 IMP_CT_FLUSH_MEM to IMP_CT_FLUSH_MEM_HAZARD	CPU_Debug (CPU_DEBUG) events for Neoverse V3
Renamed 0x0127 IMP_LS_RAR to 0x0127 IMP_LS_RAR_HAZARD	CPU_Debug (CPU_DEBUG) events for Neoverse V3
Renamed 0x0128 IMP_LS_RAW to 0x0127 IMP_LS_RAW_HAZARD	CPU_Debug (CPU_DEBUG) events for Neoverse V3

Table 3: Issue 0100-01

Change	Location
First release for all revisions of the product.	-

Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Convention	Use
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.

We recommend the following. If you do not follow these recommendations your system might not work.

Your system requires the following. If you do not follow these requirements your system will not work.

You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.

This information is important and needs your attention.

This information might help you perform a task in an easier, better, or faster way.

This information reminds you of something important relating to the current content.

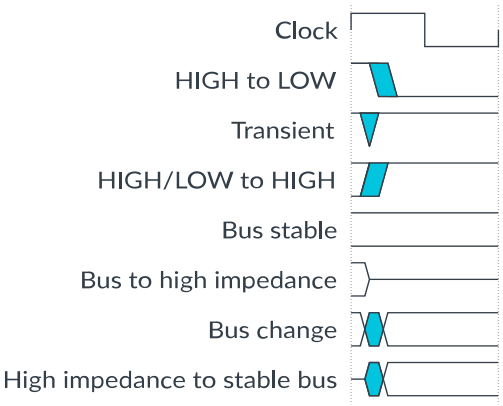
Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

Figure 1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

At the start or end of a signal name, n denotes an active-LOW signal.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Arm documents are available on developer.arm.com/documentation.

Confidential documents are only available to licensees, when logged in. Each document link in the tables below provides direct access to the online version of the document.

Arm product resources	Document ID	Confidentiality
Arm® CPU Telemetry Solution Topdown Methodology Specification	109542	Non-Confidential
Arm® Neoverse™ V3 Core Technical Reference Manual	107734	Non-Confidential
Arm® Telemetry Solution GitLab repository	–	Non-Confidential
Arm® Telemetry on Arm Developer	–	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Architecture Reference Manual for A-profile architecture	DDI 0487	Non-Confidential

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential